(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 197 34 735 A 1**

(51) Int. Cl.[6]:
G 06 F 15/18

(21) Aktenzeichen: 197 34 735.5
(22) Anmeldetag: 11. 8. 97
(43) Offenlegungstag: 18. 2. 99

(71) Anmelder:
Lory, Peter, Prof. Dr., 82319 Starnberg, DE

(74) Vertreter:
Lindner, M., Dipl.-Phys.Univ., Pat.-Anw., 81243 München

(72) Erfinder:
gleich Anmelder

(56) Entgegenhaltungen:
US 56 38 492

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

Der Inhalt dieser Schrift weicht von den am Anmeldetag eingereichten Unterlagen ab

(54) Neuronales Netz zur lernenden Vektorquantisierung und Klassifizierungsverfahren mit einem neuronalen Netz zur lernenden Vektorquantisierung

(57) Die Erfindung betrifft ein neuronales Netz zur lernenden Vektorquantisierung, mit Eingabeeinrichtungen für Eingabedatensätze und wenigstens einer Schicht aktiver Neuronen, die mit den Eingabeeinrichtungen verbunden und denen Referenzdatensätze zugeordnet sind, um zur Klassifizierung eines Eingabedatensatzes durch Vergleich mit allen Referenzdatensätzen dasjenige Neuron der Schicht aktiver Neuronen, dessen Referenzdatensatz dem Eingabedatensatz am ähnlichsten ist, zu einer Ausgabe an Ausgabeeinrichtungen zu veranlassen. Bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz und den Referenzdatensätzen ist bei zumindest einer seiner Komponenten eine vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, zugrunde gelegt. Weiterhin betrifft die Erfindung ein Klassifizierungsverfahren mit einem solchen neuronalen Netz.

DE 197 34 735 A 1

DE 197 34 735 A 1

Beschreibung

Die Erfindung betrifft ein neuronales Netz zur lernenden Vektorquantisierung nach dem Oberbegriff des Anspruchs 1 sowie ein Klassifizierungsverfahren mit einem neuronalen Netz zur lernenden Vektorquantisierung nach dem Oberbegriff des Anspruchs 20.

Bei neuronalen Netzen (NN) die häufig auch als künstliche neuronale Netze (KNN) oder "artificial neural networks" (ANN) bezeichnet werden, handelt es sich um informationsverarbeitende Systeme, die aus einer großen Anzahl einfacher Einheiten (Zellen, Neuronen) bestehen, die sich Information in Form der Aktivierung der Einheiten über gerichtete Verbindungen ("connections" "links") zusenden.

Den neuronalen Netzen liegt ihre grobe Analogie zu Säugetiergehirnen zu Grunde, bei denen Informationsverarbeitung durch sehr viele Nervenzellen stattfindet, die im Verhältnis zum Gesamtsystem sehr einfach sind und die den Grad ihrer Erregung über Nervenfasern an andere Nervenzellen weiterleiten. Im Unterschied zur Biologie und Neurologie wird bei technischen neuronalen Netzen ein Modell zu Grunde gelegt, das eine sehr grobe Verallgemeinerung darstellt. Die sich damit ergebenden Netze sind auch keine Neuronen-Netze, sondern nur "neuronale", d. h. neuronenähnliche Netze.

Neben aktuellen Bemühungen, solche neuronale Netze auch biologisch, biochemisch und chemisch zu simulieren, gibt es seit langem Ansätze von technischen Simulationen. Dabei lassen sich im Unterschied zur traditionellen Unterteilung von Rechnersystemen in die Maschine (Hardware) und in die Algorithmen, die darauf ausgeführt werden (Software), bei den neuronalen Netzen die beiden Aspekte nicht streng voneinander trennen. Ähnlich wie bei den systolischen Feldern biologischer Neuronen sind Hardwarearchitektur und Funktionsalgorithmus als Ganzes zu sehen. Die Hardwarearchitektur implementiert dabei den Algorithmus, und die Programmierung als Anpassung des allgemeinen Algorithmus an eine spezielle Aufgabe erfolgt dynamisch durch die Eingabe von Trainingsmustern.

Das Grundmodell eines Neurons stützt sich im wesentlichen auf die Vereinfachungen von McCulloch und Pitts (in W. S. McCulloch, W. H. Pitts: "A Logical Calculus of the Ideas Imminent in Nervous Activity"; Bulletin of Mathematical Biophysics (1943), Vol. 5, S. 115 bis 133). Hier wurde ein Neuron als eine Art Addierer mit Schwellenwert betrachtet. Die Verbindungen (Synapsen) eines Neurons nehmen Aktivierungen x_i mit bestimmten Stärken w_i von anderen Neuronen auf, summieren diese und lassen dann am Ausgang y (Axon) des Neurons eine Aktivität entstehen, sofern die Summe vorher einen Schwellenwert S überschritten hat.

Für weitere Einzelheiten zu neuronalen Netzen, wie sie bei der vorliegenden Erfindung zu verstehen und einzusetzen sind, und ihre detaillierten Hintergründe und Realisierungsmöglichkeiten wird auf die Bücher "Simulation Neuronaler Netze" von Andreas Zell, Addison-Wesley Publishing Company (Addison-Wesley (Deutschland) GmbH) 1994, 1. unveränderter Nachdruck 1996, und "Neuronale Netze" von Rüdiger Brause, B. G. Teubner, Stuttgart 1995, 2. überarbeitete und erweiterte Auflage, sowie dort angegebene weitere Literatur verwiesen, die zur Vermeidung umfangreicher Darstellungen im einzelnen allesamt durch diese Bezugnahme vollinhaltlich in die vorliegenden Unterlagen aufgenommen sind.

Ein wesentliches Element der künstlichen neuronalen Netze ist ihre bereits erwähnte Lernfähigkeit, d. h. die Fähigkeit, eine Aufgabe, wie beispielsweise ein Klassifizierungsproblem, selbständig aus Trainingsbeispielen zu lernen, ohne daß das neuronale Netz dazu explizit programmiert werden muß. Für Klassifizierungsprobleme besonders geeignete Versionen neuronaler Netze sind solche, die zur lernenden Vektorquantisierung ("learning vector quantization"; LVQ) ausgelegt sind. LVQ geht zurück auf T. Kohonen, der hierzu zahlreiche Publikationen verfaßte (siehe Literaturlisten in den beiden oben genannten Büchern und T. Kohonen, "Self-Organizing Maps", 2. Auflage, Springer-Verlag, Berlin Heidelberg New York, 1997).

Insbesondere gehört der vollständige Inhalt der drei vorstehend genannten Bücher und der darin angegebenen und zitierten weiteren Literatur zum Offenbarungsgehalt der vorliegenden unterlagen. Die vorliegende Erfindung betrifft daher die gesamten Ausführungsmöglichkeiten, die der Fachmann den angegebenen Publikationen entnehmen kann.

Bei neuronalen LVQ-Netzen (siehe hier insbesondere "Simulation Neuronaler Netze" von Andreas Zell, S. 171 bis 178, a.a.O.) und "Neuronale Netze" von Rüdiger Brause (S. 140 bis 147, a.a.O.) handelt es sich um sogenannte einschichtige "feedforward"-Netze, also solche, bei denen die einzelnen formalen Neuronen oder ganze Gruppen von Neuronen (Schicht) oder allgemein das Netz ohne Rückkopplung ("feedforward") Informationen verarbeiten/-t. Die Bezeichnung "ohne Rückkopplung" ist bei "feedforward"-Netzen allerdings normalerweise, was vorliegend zu Grunde gelegt wird, nur auf die Hauptflußrichtung der Signale bezogen. Beispielsweise werden weder die Auswirkungen neuronaler Ausgangssignale oder die von Fehlersignalen nachfolgender Schichten auf das Lernen von Gewichten, noch die Wechselwirkungen der Einheiten innerhalb einer Schicht bei der LVQ als Rückkopplung angesehen.

Solche einschichtigen neuronalen Netze haben somit eine auch als verdeckte Schicht ("hidden layer") zu bezeichnende Schicht aktiver Neuronen (eventuell mit vorgeschalteten Eingabeneuronen ohne Informationsverarbeitung), die oft auch als Kohonen-Neuronen bezeichnet werden. Ihre Gewichte werden durch ein einfaches Lernverfahren bestimmt, welches bei LVQ nur die Eingabe und eine Klasseninformation benutzt. LVQ ist i. d. R. ein überwachtes Lernverfahren, bei dem zu jedem Eingabedatensatz (Eingabevektor) bekannt sein muß, zu welcher Klasse (Kategorie) er gehört.

Bei der Klassifikation oder allgemein Musterkennung realisiert ein neuronales Netz eine Funktion

$$y = \varphi(X) \quad (1)$$

Dabei ist die Eingabe oder der Input

$$X = (x_1, x_2, \ldots, x_n) \quad (2)$$

ein sogenannter Muster- oder Eingabevektor oder allgemein ein Muster- oder Eingabedatensatz, und die Ausgabe oder der Output y ist aus der Menge der Klassen $C_1, C_2, \ldots, C_k$:

$y \in C = (C_1, C_2, \ldots, C_k)$ (3)

Bei der lernenden Vektorquantisierung gibt es eine (ungeordnete) Menge von sogenannten Codebuchvektoren W ("Codebook-Vektoren"; Vektoren mit freien Parametern; Referenzvektoren oder allgemein Referenzdatensätze), die für die angestrebte Klassifikation oder Quantisierung jeweils wie ein Klassenprototyp eine ganze Klasse repräsentieren und so im Eingaberaum der Eingabedatensätze x verteilt werden sollen, daß sie ihn und die durch die Funktion φ definierte Klasseneinteilung möglichst gut abdecken. Üblicherweise gibt es mehrere Codebuchvektoren W für jede Klasse von Eingabevektoren.

Das Neuronale Netz "lernt" die Funktion φ in der Trainingsphase, indem ihm eine Serie von typischen Beispielen präsentiert wird. In der anschließenden Simulationsphase kann das neuronale Netz dann zur Klassifikation verwendet werden.

Ein neuer oder bereits trainierter Eingabevektor wird parallel mit allen Referenzvektoren verglichen, und derjenige Vektor, der ihm am ähnlichsten ist, gibt seine Klasse an. Beispielsweise kann ein Codebuchvektor der Gewichtsvektor $W_j = (w_{1j}, \ldots, w_{nj})$ eines Kohonen-Neurons j sein, das von den n Eingabeneuronen seine Eingabe erhält. Existieren keine Eingabeneuronen, so wird die Eingabe von außen über allgemeine Eingabeeinrichtungen dazu verwendet.

In der **Fig.** 1 ist allgemein die Netzstruktur der lernenden Vektorquantisierung (LVQ) dargestellt, wie sie einer Ausführung der vorliegenden Erfindung zu Grunde liegt. Jedes Neuron der Eingabeschicht ist vollständig mit jedem Neuron der LVQ-Schicht verbunden. Der Codebuchvektor von Neuron j ist mit W_j bezeichnet. In zahlreichen Anwendungsfällen gibt es mehrere Codebuchvektoren (Referenzvektoren) W für jede Klasse von Eingabevektoren, wie in der **Fig.** 1 angedeutet ist. Ein Eingabevektor oder -datensatz X wird parallel mit allen Referenzvektoren oder -datensätzen $W_1, W_2, \ldots, W_m$ verglichen. Derjenige Referenzvektor W_j, der dem Eingabevektor X am ähnlichsten, wie z. B. am nächsten ist, ist der sogenannte "Gewinner" und veranlaßt das entsprechende Neuron j in der LVQ-Neuronenschicht zu "feuern". Da diese Neuronen in die Klassen $C_1, C_2, \ldots, C_k$ eingeteilt sind oder diese Klassen repräsentieren, ist damit der Eingabevektor oder -datensatz X klassifiziert.

Die Vorteile solcher neuronaler Netze sind Lernfähigkeit, Parallelität, verteilte Wissensrepräsentation, höhere Fehlertoleranz gegenüber herkömmlichen Algorithmen, Assoziative Speicherung von Information, Robustheit gegen Störungen oder verrauschte Daten, Default-Werte und spontane Generalisierung sowie aktive Repräsentation. Dennoch weisen neuronale Netze auch Nachteile auf, insbesondere dadurch, daß ein Wissenserwerb nur durch Lernen möglich ist, und daß dieses Lernen relativ langsam ist.

In zahlreichen Anwendungen sind jedoch nicht nur typische Beispiele bekannt, sondern auch gewisse Bedingungen, die bei der Klassifizierung erfüllt sein sollen. Als Stand der Technik sind hier die Publikationen N. P. Archer, S. Wang: "Application of the Back Propagation Neural Network Algorithm with Monotonicity Constraints for Two-Group Classification Problems", Decision Sciences, 24 (1993), S. 60 bis 75, und Y. S. Abu-Mustafa: "Learning from Hints", Journal of Complexity, 10 (1994), S. 165 bis 178 (insbesondere S. 170), zu nennen.

Diese Veröffentlichungen betreffen allerdings keine LVQ-Netze. Die Arbeit von Archer und Wang bezieht sich auf Backpropagation (Rückwärtsausbreitung), während die Darstellung von Abu-Mustafa allgemeiner angelegt ist. Beide Arbeiten zielen jedoch ab auf die Approximation einer monotonen Funktion.

Bei dem Verfahren nach Archer und Wang muß der eigentlichen Klassifikation als Filter eine Klassifikation aus der Linearen Diskriminanz-Analyse (LDA) vorgeschaltet werden. Dadurch wird den Daten ein Trend zur linearen Klassifikation aufgeprägt, der den Vorteil der neuronalen Netze, nichtlineare Effekte korrekt wiederzugeben, nicht voll wirksam werden läßt. Außerdem steckt in der Veränderung der gegebenen Daten eine unverständliche Willkür. Dies gilt insbesondere auch wegen der Verwendung von willkürlichen Parametern.

Das von Abu-Mustafa vorgeschlagene Trainingsverfahren besteht in einer Vergrößerung der Beispielmenge um Beispiele, die eine Monotonie reflektieren. Dadurch wird der Trainingsaufwand deutlich erhöht. Es bleibt unklar, wieviel neue Beispiele eingeschleust werden müssen, damit die Monotonie ausreichend gewährleistet wird. In einem strikten Sinne kann sie bei der Methode von Abu-Mustafa sowieso nicht garantiert werden.

Es ist daher ein Ziel der vorliegenden Erfindung, ein neuronales Netz zur lernenden Vektorquantisierung sowie ein Klassifizierungsverfahren mit einem neuronalen Netz zur lernenden Vektorquantisierung zu schaffen, wobei vorbekannte Bedingungen bei der Klassifizierung zuverlässig eingehalten werden.

Dieses Ziel wird vorrichtungsmäßig mit einem neuronalen Netz zur lernenden Vektorquantisierung nach dem Anspruch 1 und verfahrensmäßig mit einem Klassifizierungsverfahren für Eingabedatensätze mit einem neuronalen Netz zur lernenden Vektorquantisierung nach Anspruch 20 erreicht.

Erfindungsgemäß enthält ein neuronales Netz zur lernenden Vektorquantisierung Eingabeeinrichtungen für Eingabedatensätze und wenigstens eine Schicht aktiver Neuronen, die mit den Eingabeeinrichtungen verbunden und denen Referenzdatensätze zugeordnet sind, um zur Klassifizierung eines Eingabedatensatzes durch Vergleich mit allen Referenzdatensätzen dasjenige Neuron der Schicht aktiver Neuronen, dessen Referenzdatensatz dem Eingabedatensatz am ähnlichsten ist, zu einer Ausgabe an Ausgabeeinrichtungen zu veranlassen. Zur Einhaltung vorbekannter Bedingungen ist dabei vorgesehen, daß bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und den Referenzdatensätzen W bei zumindest einer seiner Komponenten x_i eine vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, zu Grunde gelegt ist.

Die vorliegende Erfindung gibt für den Netztyp der lernenden Vektorquantisierung (LVQ) eine Technik an, welche garantiert, daß die Funktion φ im strikten Sinn eine Eigenschaft, wie z. B. eine Monotonieeigenschaft, besitzt. Für die Einspeisung eines derartigen prozeduralen Wissens in das neuronale Netz sind keine standardmäßigen Lösungen bekannt. Zwar wird in T. Kohonen: "6. Learning Vector Quantization" (a.a.O.) und Andreas Zell: "Simulation Neuronaler Netze" (a.a.O.) die prinzipielle Möglichkeit der Verwendung eines anderen als des euklidischen Abstands als Ähnlichkeitskriterium bei LVQ erwähnt, jedoch wird für den exemplarisch hier genannten praktisch wichtigen Fall der Monotonie kein geeigneter Abstand angegeben.

Bei dem durch die Erfindung geschaffenen neuronalen Netz werden die geforderten Eigenschaften, wie beispielsweise

Monotonien, bei der Klassifikation strikt eingehalten. Vorteilhaft ist darüber hinaus, daß der Trainingsaufwand deutlich reduziert wird, da die Eigenschaften, wie z. B. Monotonieeigenschaften, nicht mehr durch Beispiele trainiert werden müssen. Dies ist besonders wichtig bei Problemen mit ungünstiger Datensituation.

Vorzugsweise ist bei dem neuronales Netz nach der Erfindung weiter vorgesehen, daß die Eingabeeinrichtungen eine Eingabeneuronenschicht enthalten, deren Eingabeneuronen jeweils vollständig mit jedem Neuron der Schicht aktiver Neuronen verbunden sind, wodurch eine besonders schnelle Datenverarbeitung ermöglicht wird. Gemäß einer bevorzugten Weiterbildung davon kann vorgesehen sein, daß jedem Eingabeneuron der Eingabeneuronenschicht eine Komponente x_i eines Eingabedatensatzes X zugeordnet ist.

Mit Vorteil sind bei einem neuronalen Netz der Erfindung die Neuronen einer Schicht aktiver Neuronen in Ausgabeklassen eingeteilt. Dabei kann weiterhin vorgesehen sein, daß jeder Ausgabeklasse wenigstens ein Neuron und insbesondere eine Vielzahl von Neuronen einer Schicht aktiver Neuronen zugeordnet ist.

Eine andere bevorzugte Ausgestaltung der Erfindung sieht vor, daß die Ausgabeeinrichtungen eine Ausgabeneuronenschicht enthalten, deren Ausgabeneuronen jeweils eine Ausgabeklasse repräsentieren und in Abhängigkeit von dem ermittelten Neuron der vorhergehenden Schicht aktiver Neuronen aktivierbar sind.

Bei den bisher geschilderten weiterführenden Ausgestaltungen hinsichtlich Ausgabeklassen kann die vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in zumindest wenigstens einer Richtung oder Symmetrie, für zumindest eine Ausgabeklasse gelten.

Vorzugsweise erfolgt bei einem neuronalen Netz, wie es durch die Erfindung geschaffen wurde, die Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und jedem der Referenzdatensätze W in der Weise, daß derjenige Referenzdatensatz W bestimmt wird, der den geringsten Abstand, insbesondere den geringsten euklidischen Abstand von dem Eingabedatensatz X aufweist. Ein solches Ähnlichkeitskriterium kann mit geringem Rechenaufwand und zuverlässig angewandt werden. Dies ermöglicht wiederum ein schnelles Training und eine schnelle Klassifizierung von Musterdatensätzen, deren Klassenzugehörigkeit ermittelt werden soll, sowie eine hohe Reproduzierbarkeit der Anwendung.

Eine besonders bevorzugte Ausgestaltung des erfindungsgemäßen neuronalen Netzes ist zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und jedem der Referenzdatensätze W dazu ausgelegt, zu denjenigen Referenzdatensätzen W, deren Klasse wenigstens eine vorgegebene Eigenschaft, wie z. B. eine Monotonieeigenschaft, besitzt, eine Punktmenge zu bilden, die aus einem Referenzdatensatz und denjenigen Datensätzen gebildet ist, die von dem Referenzdatensatz aus in der/den Richtung(en) der einen oder mehreren vorgegebenen Eigenschaft(en), wie z. B. Monotonieeigenschaft(en), liegen, und anhand derjenigen Punktmenge, von der der Eingabedatensatz X den geringsten Abstand, insbesondere den geringsten euklidischen Abstand, aufweist, den zugehörigen Referenzdatensatz W als denjenigen festzulegen, der dem Eingabedatensatz X am ähnlichsten ist. Wenn man die Referenzdatensätze W als Punkte in einem mehrdimensionalen Raum betrachtet, dann können die zugehörigen Punktmengen die von einschließlich dem Punkt eines Referenzdatensatzes W ausgehenden Halbgeraden, Teilebenen oder entsprechenden höherdimensionalen Gebilde sein, die den Punkt des jeweiligen Referenzdatensatzes W enthalten und in der/den Monotonierichtung(en) verlaufen. Anders ausgedrückt, sind somit diejenigen Datensätze, die von dem Referenzdatensatz aus in einer Monotonierichtung oder mehreren Monotonierichtungen liegen, diejenigen, die eine oder mehrere von einem Referenzdatensatz W ausgehende Halbgerade(n), Teilebene(n) oder entsprechende höherdimensionale Gebilde bilden.

In Anbetracht der praktischen Bedeutung der Monotonie bei Klassifizierungsaufgaben ist bei einer besonders bevorzugten Weiterbildung des erfindungsgemäßen neuronalen Netzes vorgesehen, daß zur Klassifizierung eines Eingabedatensatzes X bei zumindest einer seiner Komponenten x_i wenigstens ein Gültigkeitsbereich einer vorgegebenen Eigenschaft, wie z. B. einer Monotonieeigenschaft, zu Grunde gelegt ist.

Im Rahmen der Erfindung ist es ferner bevorzugt, wenn ein Gültigkeitsbereich einer vorgegebenen Eigenschaft, wie z. B. einer Monotonieeigenschaft, der Komponente x_i des Eingabedatensatzes X die positive oder negative i-Richtung der Komponente x_i des Eingabedatensatzes X ist.

Nachfolgend werden einige besonders bevorzugte Realisierungen im Rahmen der Erfindung zur Aufprägung einer Monotonieeigenschaft bei der Klassifizierung oder Mustererkennung angegeben.

Bei einer mit Vorzug verwendeten Variante des neuronalen Netzes nach der Erfindung ist vorgesehen, daß es bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und jedem der Referenzdatensätze W die der vorgegebenen oder vorbekannten Eigenschaft, wie z. B. einer Monotonieeigenschaft, der Klassifizierung entsprechende Komponente x_i des Eingabedatensatzes X unberücksichtigt läßt, wenn der Eingabedatensatz in dem entsprechenden Gültigkeitsbereich der vorgegebenen Eigenschaft, wie z. B. einer Monotonieeigenschaft, der Klassifizierung liegt und gegenüber dem Referenzdatensatz W eine dieser Eigenschaft entsprechende Lage hat (vgl. z. B. **Fig.** 2 und die Beschreibung dazu weiter unten).

Alternativ ist ein erfindungsgemäßes neuronales Netz auch dadurch zu erreichen, daß es den Abstand zwischen einem Eingabedatensatz X und jedem der Referenzdatensätze W bestimmt, indem zu allen Referenzdatensätzen W, welche zu Klassen mit einer oder mehreren vorgegebenen Eigenschaften, wie z. B. Monotonieeigenschaft(en), gehören, ausgehend von jedem dieser Referenzdatensätze W in Richtung der einen oder mehreren vorgegebenen Eigenschaften, wie z. B. Monotonieeigenschaft(en), eine zugehörige Punktmenge gebildet wird und dann der Abstand des Eingabedatensatzes X von dieser Punktmenge bestimmt wird, soweit diese in dem Gültigkeitsbereich oder den Gültigkeitsbereichen der vorgegebenen Eigenschaft(en) liegt.

Für den speziellen Fall einer Monotonieeigenschaft einer Komponente x_i der Eingabedatensätze X in positiver i-Richtung ist es bevorzugt, wenn bei dem erfindungsgemäßen neuronalen Netz der Abstand zwischen einem Eingabedatensatz X und den Referenzdatensätzen W gemäß

$$d(X,W) = \begin{cases} \sqrt{\sum_{\substack{v=1 \\ v \neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i > w_i \\ \|X - W\| = \sqrt{\sum_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \leq w_i \end{cases} \tag{6}$$

bestimmt wird, wobei $X = (x_1, \ldots, x_n)$ und $W = (w_1, \ldots, w_n)$.

Analog gilt für den Fall, daß eine Monotonieeigenschaft einer Komponente x_i der Eingabedatensätze X in negativer i-Richtung vorliegt, daß das neuronale Netz den Abstand zwischen den Eingabedatensätzen X und den Referenzdatensätzen W vorzugsweise gemäß

$$d(X,W) = \begin{cases} \sqrt{\sum_{\substack{v=1 \\ v \neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i < w_i \\ \|X - W\| = \sqrt{\sum_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \geq w_i \end{cases} \tag{7}$$

bestimmt wird, wobei $X = (x_1, \ldots, x_n)$ und $W = (w_1, \ldots, w_n)$.

Eine solche Berechnung des Abstands bedeutet in vorteilhafter Weise keine Erhöhung des erforderlichen Rechenaufwandes. Zwar kommt eine Abfrage hinzu, jedoch erübrigt sich in statistisch 50% der Fälle eine Quadrierung und eine Addition sowie eine Differenzbildung.

Weiterhin kann zur vorteilhaften Realisierung der Erfindung vorgesehen sein, daß die Neuronen der Schicht aktiver Neuronen jeweils eine Prozessoreinheit und einen lokalen Speicher zur Speicherung eines Referenzdatensatzes W enthalten, daß die Schicht durch eine Parallelanordnung der Prozessoreinheiten mit ihren zugeordneten Speichern gebildet ist, und daß jede Prozessoreinheit zur Durchführung des Vergleichs des eingegebenen Eingabedatensatzes X mit dem Referenzdatensatz W in dem zugehörigen Speicher ausgelegt ist.

Besonders bevorzugt ist eine erfindungsgemäße Ausgestaltung des neuronalen Netzes, bei der die Schicht aktiver Neuronen zum parallelen Vergleich eines Eingabedatensatzes X mit allen Referenzdatensätzen W ausgelegt ist.

Ferner liegt es ebenfalls im Rahmen der Erfindung, wenn, wie bei einer anderen Ausführung des neuronalen Netzes, die Neuronen der Schicht aktiver Neuronen zumindest durch eine zentrale Verarbeitungseinheit und einen zentralen Speicher zum Speichern aller Referenzdatensatzes W gebildet sind, und wenn die zentrale Verarbeitungseinheit zum Durchführen des Vergleichs des eingegebenen Eingabedatensatzes X mit jedem Referenzdatensatz W in dem zentralen Speicher in paralleler oder bezüglich jedes Referenzdatensatzes W unmittelbar serieller oder bezüglich jeder Komponente x_i des Eingabedatensatzes X unmittelbar serieller Weise ausgelegt ist. Solche unmittelbaren seriellen Simulationen einer parallelen Verarbeitung können auch als quasiparallele Verarbeitungen angesehen werden, wobei eventuelle zeitliche Nachteile durch entsprechend hohe Rechnerleistung, wie insbesondere Taktgeschwindigkeit oder Wortbreite, kompensiert werden können.

Die Auswahl einer geeigneten Hardwarebasis hängt von den Möglichkeiten, wie u. a. Finanzen, von dem Einsatzbedarf, wie z. B. mobile Rechner oder wenige oder zahlreiche individuelle Systeme, und von dem zu bearbeitenden Problem ab. Die Erfindung ist jedoch nicht auf einzelne Rechnerarchitekturen oder Softwareplattformen beschränkt, sondern kann, die Implementierung eines neuronalen Netzes durch oder auf einem Computer von im wesentlichen beliebiger Bauart vorausgesetzt, im Rahmen aller bekannten Systeme umgesetzt und betrieben werden.

Wie bereits weiter oben angegeben wurde, können die Eingabedatensätze X durch Eingabevektoren und die Referenzdatensätze W durch Referenzvektoren gebildet sein.

Bei dem erfindungsgemäßen Klassifizierungsverfahren für Eingabedatensätze mit einem neuronalen Netz zur lernenden Vektorquantisierung werden Eingabedatensätze über Eingabeeinrichtungen einer Ähnlichkeitsbestimmung bezüglich Referenzdatensätzen unterzogen, die jeweils einem Neuron einer Schicht aktiver Neuronen zugeordnet sind, und es wird zur Klassifizierung durch Vergleichen eines Eingabedatensatzes mit allen Referenzdatensätzen dasjenige Neuron der Schicht aktiver Neuronen, dessen Referenzdatensatz dem Eingabedatensatz am ähnlichsten ist, eine Ausgabe an Ausgabeeinrichtungen veranlaßt, wobei bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und den Referenzdatensätzen W bei zumindest einer seiner Komponenten x_i eine vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, zu Grunde gelegt wird.

Die bevorzugten und mit Vorteil anwendbaren Fortbildungen dieses Klassifizierungsverfahrens im Rahmen der Erfindung sind analog zu den erfindungsgemäß weiterbildenden vorrichtungsmäßigen Ausführungen, die weiter oben angegeben und behandelt sind.

Weitere vorteilhafte Ausgestaltungen ergeben sich aus den abhängigen Ansprüchen und deren Kombinationen.

Die Erfindung wird nachfolgend anhand konkreter Ausführungsbeispiele näher beschrieben, ohne daß damit eine Beschränkung der Erfindung auf die angegebenen Varianten verbunden sein soll. Der gesamte Umfang der Erfindung wird durch den vollständigen Offenbarungsgehalt dieser Beschreibung und der Ansprüche bestimmt, wobei sämtliche Variationen, Modifikationen und Substitutionen durch die Erfindung abgedeckt sind, die der Fachmann ohne weiteres aus der Gesamtheit dieser Unterlagen zu erkennen vermag. Dabei sind in deren Offenbarungsumfang durch die Bezugnahme im vorstehenden einleitenden Teil dieser Beschreibung insbesondere auch die dort angegebenen und teilweise zitierten Druckschriften und Bücher vollinhaltlich enthalten.

Das grundsätzliche Prinzip eines neuronalen LVQ-Netzwerkes ist in der **Fig.** 1 veranschaulicht, auf die bereits im voranstehenden einleitenden Teil dieser Beschreibung eingegangen wurde.

Die **Fig.** 2 zeigt eine schematische Darstellung zur Erläuterung der Abstandsbestimmung bei einem ersten Ausführungsbeispiel eines neuronalen LVQ-Netzes mit Monotonie in positiver 2-Richtung.

Die **Fig.** 3 ist eine Darstellung der Klassengrenze bei einem neuronalen LVQ-Netz mit zwei Klassen ohne Monotonieeigenschaften.

Die **Fig.** 4 illustriert die Klassengrenze, die sich durch Anwendung der Formeln (6) und (7) ergibt, bei einem neuronalen LVQ-Netz mit zwei Klassen, wobei die Klasse C_{ja} monoton in positiver 2-Richtung und die Klasse C_{nein} monoton in negativer 2-Richtung ist.

Die **Fig.** 5a und 5b zeigen ein erstes Beispiel einer Simulation des erfindungsgemäßen neuronalen Netzes bzw. des entsprechenden Klassifizierungsverfahrens.

Die **Fig.** 6a und 6b zeigen ein zweites Beispiel einer Simulation des erfindungsgemäßen neuronalen Netzes bzw. des entsprechenden Klassifizierungsverfahrens.

Bei neuronalen Netzen wird im Rahmen der lernenden Vektorquantisierung eine Ähnlichkeitsbestimmung durchgeführt, die ohne Beschränkung auf bestimmte Ähnlichkeitskriterien nachfolgend anhand einer durch einen Abstand bestimmten Ähnlichkeit erläutert wird.

Um zu präzisieren, was unter dem am nächsten liegenden Referenzdatensatz oder -vektor (Codebuchvektor) verstanden werden soll, ist die Definition eines Abstands nötig. Eine häufig verwendete Form dafür ist die euklidische Norm

$$\|X\| := \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} \qquad (8)$$

wobei $X = (x_1, \ldots, x_n)$.

Die Größe $d(X,W_j) = \|X-W_j\|$ stellt dann den bekannten euklidischen Abstand zwischen dem Eingabevektor X und dem Referenzvektor W_j dar. Der Gewinner W_c unter den Referenzvektoren ist also definiert durch

$$\|X - W_c\| = \min_j (\|X - W_j\|) \quad (9)$$

wobei $X = (x_1, \ldots, x_n)$ und $W_j = (w_1, \ldots, w_{nj})$.

Als Trainingsmethoden zur Ermittlung von Referenzfaktoren, welche die Klasseneinteilung gut abdecken, sind verschiedene Verfahren in Gebrauch: LVQ1, LVQ2, LVQ2.1, LVQ3, OLVQ1. Man vergleiche dazu T. Kohonen: "Self-Organizing Maps" ("6. Learning Vector Quantization", S. 203ff., a.a.O.) und Andreas Zell: "Simulation Neuronaler Netze" (S. 172ff., a.a.O.).

Im folgenden sollen nun einige (evtl. auch alle) der Klassen aus $\{C_1, C_2, \ldots, C_k\}$ eine a priori bekannte Monotonieeigenschaft besitzen. Dabei ist eine Klasse C monoton in positiver i-Richtung, falls gilt:
Ist der Eingabevektor $X_0 = (x_{10}, x_{20}, \ldots, x_{n0})$ aus der Klasse C, so ist auch jeder andere Eingabevektor $X = (x_1, x_2, \ldots, x_n)$ mit der Eigenschaft $x_i > x_{i0}$ und $x_j = x_{j0}$ für $j \neq i$ aus der Klasse C.

Entsprechend ist eine Klasse C monoton in negativer i-Richtung, falls gilt:
Ist der Eingabevektor $X_0 = (x_{10}, x_0, \ldots, x_{n0})$ aus der Klasse C, so ist auch jeder andere Eingabevektor $X = (x_1, x_2, \ldots, x_n)$ mit der Eigenschaft $x_i < x_{i0}$ und $x_j = x_{j0}$ für $j \neq i$ aus der Klasse C.

Im Falle einer exemplarisch angegebenen Anwendung zur Kreditwürdigkeitsprüfung bei Privatkunden gäbe es die zwei Klassen C_{ja} und C_{nein}, welche für die Annahme bzw. Ablehnung des Kreditantrags stehen. Falls die Komponente x_1 das Gehalt des Kreditbewerbers bezeichnet, so ist es sicherlich wünschenswert und sinnvoll, daß die Klasse C_{ja} monoton in positiver 1-Richtung und die Klasse C_{nein} monoton in negativer 1-Richtung ist. Derartigen Monotonie-Eigenschaften kann durch eine geeignete Wahl des Abstandes zwischen Eingabe- und Codebuchvektor Rechnung getragen werden:
Für das vorstehend aufgezeigte Ausführungsbeispiel ist, wenn die Klasse C monoton in positiver i-Richtung ist, der Abstand zwischen dem Eingabevektor $X = (x_1, x_2, \ldots, x_n)$ und dem zur Klasse C gehörigen Codebuchvektor oder allgemein Referenzdatensatz $W = (w_1, w_2, \ldots, w_n)$ bestimmt durch

$$d(X,W) = \begin{cases} \sqrt{\sum\limits_{\substack{v=1 \\ v \neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i > w_i \\ \|X - W\| = \sqrt{\sum\limits_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \leq w_i \end{cases} \qquad (6)$$

Ist hingegen die Klasse C monoton in negativer i-Richtung, so wird der entsprechende Abstand bestimmt durch

$$d(X,W) = \begin{cases} \sqrt{\sum\limits_{\substack{v=1 \\ v \neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i < w_i \\ \|X - W\| = \sqrt{\sum\limits_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \geq w_i \end{cases} \qquad (7)$$

Im Falle n=2 läßt sich die Bestimmung veranschaulichen, wie in der **Fig.** 2 gezeigt ist.

Eine solche Berechnung des Abstands bedeutet keine Erhöhung des erforderlichen Rechenaufwandes. Zwar kommt eine Abfrage hinsichtlich der Größe von x_i bezüglich w_i hinzu, jedoch erübrigt sich in statistisch 50% der Fälle eine Differenzbildung, eine Quadrierung und eine Addition.

Bei einer besonders bevorzugten Ausgestaltung des erfindungsgemäßen neuronalen Netzes wird als Voraussetzung zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und jedem der Referenzdatensätze W, für deren Klasse eine Monotonieeigenschaft gültig ist, zunächst zu jedem dieser Referenzdatensätze W eine Punktmenge gebildet, die aus einem Referenzdatensatz und denjenigen Datensätzen besteht, die von dem Referenzdatensatz aus in einer Monotonierichtung oder mehreren Monotonierichtungen liegen. Die damit verbundene bevorzugte allgemeine Art einer Abstandsbestimmung zur Ermittlung der Ähnlichkeit zwischen Referenzdatensätzen W und Eingabedatensätzen X ist in der **Fig.** 2 verdeutlicht. Die Punktmenge zum Referenzdatensatz W ist beispielsweise durch die von dem den letzteren repräsentierenden Punkt ausgehende Halbgerade parallel zur 2-Richtung gebildet (punktierte Linie in der **Fig.** 2). Als Maß für die Ähnlichkeit eines Eingabedatensatzes X zu einem Referenzdatensatz W wird der z. B. euklidische Abstand zwischen dem Eingabedatensatz X und der Punktmenge ermittelt, der der Referenzdatensatz W angehört.

Als Abstand eines Punktes von einer Punktmenge wird der kürzeste Abstand des Punktes von einem Punkt der Punktmenge herangezogen. D. h., daß gemäß der **Fig.** 2 als der Abstand des Eingabedatensatzes X_1 vom Referenzdatensatz W der Abstand des Eingabedatensatzes X_1 von der Halbgeraden erhalten wird, die vom Referenzdatensatz W aus in positiver 2-Richtung (i=2) verläuft. Dies liegt daran, daß eine Monotonie in positiver 2-Richtung vorgegeben und der Wert der 2-Komponente (i=2) des Eingabedatensatzes X_1 größer als der Wert der 2-Komponente (i=2) des Referenzdatensatzes W ist. Die Größe des Abstandes zwischen dem Eingabedatensatz X_1 und dem Referenzdatensatz W bemißt sich längs des Lotes vom Eingabedatensatz X_1 auf die Halbgerade (punktierte Linie in der **Fig.** 2), die in der positiven 2-Richtung vom Referenzdatensatz W ausgeht.

Statt einer Monotonie in positiver 2-Richtung, d. h. parallel zur x_2-Achse in deren positiver Richtung verlaufend, kann beispielsweise eine Monotonie entsprechend relativ zu jeder anderen vorhandenen positiven oder negativen i-Richtung a priori vorgegeben sein. In diesem allgemeineren Fall wird dann als Abstand zwischen einem Eingabedatensatz X und einem Referenzdatensatz W derjenige Abstand festgelegt, den der Eingabedatensatz X von der Halbgeraden hat, die vom Referenzdatensatz W aus in entsprechend der Monotoniebedingung positiver oder negativer i-Richtung verläuft. Der Abstandswert wird somit durch die Ermittlung des Abstandes des Eingabedatensatzes X_1 von der durch die genannte Halbgerade (punktierte Linie in der **Fig.** 2) gebildeten Punktmenge erhalten.

Wenn man die Referenzdatensätze W als Punkte in einem mehrdimensionalen Raum betrachtet, dann können die zugehörigen Punktmengen die von einschließlich dem Punkt eines Referenzdatensatzes W ausgehenden Halbgeraden, Teilebenen oder entsprechend höherdimensionalen Gebilde sein, die den Punkt des jeweiligen Referenzdatensatzes W enthalten und in der Monotonierichtung oder den Monotonierichtungen verlaufen.

In Abhängigkeit von der konkreten Eigenschaft, wie z. B. einer Monotonieeigenschaft, die dem neuronalen Netz aufgeprägt wurde, kann die Punktmenge aber auch durch beliebig liegende Geraden, gekrümmte Linien, Geraden- oder Kurventeile, plane oder gekrümmte Ebenen oder Ebenenausschnitte mit jeweils beliebiger, nur durch die vorgegebene Eigenschaft bestimmter Lage repräsentiert werden, wobei gemäß der hier behandelten Ausführungsform der Ähnlichkeitsbestimmung immer der Abstand eines Eingabedatensatzes X zum diesem am nächsten liegenden Punkt der Punktmenge als Maß für den Abstand dieses Eingabedatensatzes X vom zu der Punktmenge gehörenden Referenzdatensatz W gilt.

Der Referenzdatensatz W repräsentiert gleichsam die ihm zugeordnete oder von ihm ausgehende Punktmenge. Somit ist einem Eingabedatensatz X derjenige Referenzdatensatz W am ähnlichsten, von dessen Punktmenge der Eingabedatensatz X den geringsten Abstand, insbesondere den geringsten euklidischen Abstand, aufweist. Anders betrachtet kann der Sachverhalt auch so formuliert werden, daß die Datensätze der zu einem Referenzdatensatz W gehörenden Punktmenge Repräsentanten dieses Referenzdatensatzes W sind, die seine Wirkungsumgebung im Sinne der vorgegebenen Eigenschaft, wie z. B. der Monotonieeigenschaft, formen, indem diese Datensätze die Wirkungsumgebung ausdehnen und/oder verlegen. Die so definierte Wirkungsumgebung eines Referenzdatensatzes W hängt somit in Ausdehnung, Lage und Form von der vorgegebenen Eigenschaft, wie z. B. einer Monotonieeigenschaft, ab und beeinflußt den Abstand, der einem Eingabedatensatz X bezüglich des Referenzdatensatzes W zugewiesen wird, auch wenn der Abstand des Eingabedatensatzes X von dem Referenzdatensatz W tatsächlich größer ist.

Der Vollständigkeit halber wird noch auf den Fall eingegangen, der in der **Fig.** 2 exemplarisch durch den Punkt des Eingabedatensatzes X_2 repräsentiert wird. Anders als beim Eingabedatensatz X_1 liegt der Wert der 2-Komponente des Eingabedatensatzes X_2 nicht im Bereich der Werte der 2-Komponente der Datensätze, die die vom Referenzdatensatz W in positiver 2-Richtung ausgehende Halbachse bilden, so daß vom dem Eingabedatensatz X_2 entsprechenden Punkt aus auf diese Halbgerade kein Lot gefällt werden kann. Daher wird der Abstand des Eingabedatensatzes X_2 vom Referenzdatensatz W als der kürzeste z. B. euklidische Abstand zwischen den den Eingabedatensatz X_2 und den Referenzdatensatz W repräsentierenden Punkten ermittelt, wie es auch ohne eine beispielsweise vorgegebene Monotonieeigenschaft der Fall wäre. Die dem Referenzdatensatz W zugeordnete Punktmenge, die gleichsam die beispielsweise zu Grunde gelegte Monotonieeigenschaft zum Ausdruck bringt, spielt somit in diesem Fall keine Rolle.

Soweit vorstehend von Punkten gesprochen wurde, die Referenzdatensätze W und diesen auf Grund einer vorbekannten oder vorgegebenen Eigenschaft, wie z. B. einer Monotonieeigenschaft, zugeordnete Punktmengen sowie Eingabedatensätze X repräsentieren, erfolgte dies zum Zwecke der Verdeutlichung. In einem anderen Modell könnten die Referenzdatensätze W, die diesen auf Grund einer vorbekannten Eigenschaft, wie z. B. einer Monotonieeigenschaft, zugeordneten Punktmengen und die Eingabedatensätze X durch Vektoren beschrieben werden, wobei dann die vorstehenden Erläuterungen zur Punktmengenbildung und Abstandsbestimmung entsprechend gelten. Sowohl bei dem Punkt- als auch bei dem Vektormodell handelt es sich lediglich um Beschreibungen einer dem erfindungsgemäßen neuronalen Netz innewohnenden Eigenschaft. Die Modellvorstellung ermöglicht dabei das Verständnis, wie bei Ausführungsformen des erfindungsgemäßen neuronalen Netzes bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz X und den Re-

ferenzdatensätzen W zu verfahren ist, falls bei zumindest einer seiner Komponenten x_i eine vorbekannte oder vorgegebene Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, zu Grunde gelegt ist.

Nachfolgend sollen in einigen einfachen, aber charakteristischen Fällen die Grenzen zwischen den Klassen bei Verwendung der Formeln (4), (6) und (7) im Raum der Eingabevektoren veranschaulicht werden. Sei dazu n = 2, und es gebe k = 2 Klassen: $C = \{C_{ja}, C_{nein}\}$. Jede Klasse werde nur durch jeweils einen Codebook- oder Codebuchvektor, d. h. einen Referenzdatensatz repräsentiert.

In einem ersten Fall, der in der **Fig.** 3 gezeigt ist, haben sowohl C_{ja}, als auch C_{nein} keine Monotonieeigenschaft. Dann ist die Klassengrenze gerade die Mittelsenkrechte zwischen den beiden Endpunkten der Codebook-Vektoren.

In der **Fig.** 4 ist für einen zweiten Fall die Klassengrenze dargestellt. Hier ist die Klasse C_{ja} monoton in positiver 2-Richtung, und die Klasse C_{nein} ist monoton in negativer 2-Richtung. Die Klassengrenze ist zusammengesetzt aus geradenstücken und Parabeln, die dadurch erhalten werden, daß die Punkte ermittelt werden, die von den Referenzdatensätzen W_{ja} und W_{nein} bzw. den diesen zugeordneten Punktmengen (punktierte Halbgeraden in der **Fig.** 4 in positiver 2-Richtung ausgehend vom Referenzdatensatz W_{ja} und in negativer 2-Richtung ausgehend vom Referenzdatensatz W_{nein}) gleiche Abstände haben.

Im Falle von mehreren Codebook-Vektoren pro Klasse setzen sich die Grenzen aus den obigen Elementen zusammen.

Diese Beispiele sollen nun für den praktisch wichtigen Fall zweier Klassen mit gegenläufigen Monotonie-Eigenschaften weiterentwickelt werden:

Voraussetzung ist, daß eine Klasseneinteilung in zwei Klassen vorliegt: $C = \{C_{ja}, C_{nein}\}$. Die Klasse C_{ja} sei monoton in positiver i-Richtung und die Klasse C_{nein} in negativer i-Richtung.

Verwendet man dem durch die Gleichungen (6) und (7) eingeführten Abstand zwischen den Eingabevektoren oder -datensätzen einerseits und den Codebook-Vektoren oder Referenzdatensätzen andererseits, so gibt das neuronale Netz die Monotonie-Eigenschaften korrekt wieder.

Dies läßt sich auch beweisen. Ist nämlich der Eingabevektor $X_0 = (x_{10}, x_{20}, \ldots, x_{n0})$ aus der Klasse C_{ja}, so hat er zu einem Codebuch- oder Codebook-Vektor aus dieser Klasse im Vergleich zu den Abständen zu den anderen Codebook-Vektoren einen minimalen Abstand. Wegen der Gleichung (6) kann dieser Abstand nicht größer werden, wenn x_{i0} vergrößert wird. Ebenso können die Abstände von x_0 zu den Codebook-Vektoren der Klasse C_{nein} nicht kleiner werden, wenn x_{i0} vergrößert wird. Somit bleibt x_0 auch bei vergrößertem x_{i0} in der Klasse C_{ja}. Dies bedeutet die Monotonie in positiver i-Richtung. Ebenso zeigt sich die Monotonie in negativer i-Richtung.

Verwendet man also in der Simulationsphase den neuen Abstandsbegriff, so gibt das neuronale Netz die Monotonie-Eigenschaften korrekt wieder. Konsequenterweise sollte auch in der Trainingsphase derselbe Abstand verwendet werden.

Die folgenden Beispiele sind sehr einfach gewählt, um sie möglichst nachvollziehbar zu halten. Sie belegen die Wirksamkeit der neuen Methode. Sie wurden berechnet mit der MATLAB Neural Network TOOLBOX (H. Demuth, M. Beale; The MathWorks, Inc. Natick, Mass. 1994). Die darin enthaltene Abstandsberechnung wurde durch das neue erfindungsgemäße neuronale Netz bzw. das analoge Verfahren entsprechend modifiziert. Die zugehörigen Graphiken sind für das Beispiel 1 in den **Fig.** 5a und 5b und für das Beispiel 2 in den **Fig.** 6a und 6b gezeigt.

Beispiel 1

Dieses Beispiel hat als Trainingsmenge die drei Spaltenvektoren der Matrix P:

$$P = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 3 \end{pmatrix} \qquad (10)$$

Die ersten beiden Spaltenvektoren gehören zur Klasse C_{ja}, der letzte zur Klasse C_{nein} Dies wird an das System übergeben durch eine Matrix T, wobei eine Spalte $(1,0)^T$ für ja und eine Spalte $(0,1)^T$ für nein steht. Eine Matrix W1 enthält am Anfang die Startvektoren für das Training der Codebook-Vektoren oder Referenzdatensätze. Eine Matrix W2 legt hier fest, daß jede Klasse durch genau einen Referenzdatensatz oder Codebook-Vektor repräsentiert werden soll. Nach der Trainingsphase (1000 Epochen) enthalten die Zeilen von W1 die ermittelten Codebook-Vektoren.

Ohne Berücksichtigung der Monotonie sind das die Codebook-Vektoren

$$w1 = (1.5782, 1.0012), w2 = (2.9637, 2.9637) \quad (11)$$

Diese sind in der entsprechenden Graphik der **Fig.** 5a als O und ⊠ markiert, während die Vektoren der Trainingsmenge mit + und ⋇ gekennzeichnet sind. Die Symbole O und + stehen dabei für die Klasse C_{ja}, während die Symbole ⊠ und ⋇ die Klasse C_{nein} kennzeichnen. Die vom neuronalen Netz ermittelt Klassengrenze ist hier die Mittel senkrechte zwischen den beiden Codebook-Vektoren. Sie erfüllt die Forderung der Monotonie in positiver und negativer 2-Richtung natürlich nicht, wie durch Simulationen mit dem Programm "simulvq" von MATLAB Neural Network TOOLBOX (a.a.O.) belegt werden kann.

Mit Berücksichtigung der Monotonie des neuronalen Netzes bzw. des damit ausgeführten Verfahrens nach der vorliegenden Erfindung werden als Referenzdatensätze oder Codebook-Vektoren berechnet

$$w1 = (1.5781, 1.0011), w2 = (2.9725, 2.9951) \quad (12)$$

die in der **Fig.** 5b in analoger Symbolik zur **Fig.** 5a gezeigt sind.

Die vom neuronalen Netz ermittelte Klassengrenze ähnelt hier der **Fig.** 4. Die Forderung der Monotonie in positiver

und negativer 2-Richtung ist erfüllt. Dies kann wieder durch Simulationen mit dem Programm "sinulvq" von MATLAB Neural Network TOOLBOX (a.a.O.) belegt werden.

Beispiel 2

Dieses Beispiel hat als Trainingsmenge die neun Spaltenvektoren der Matrix P:

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 4 & 5 & 5 & 6 & 7 \\ 5 & 4 & 4 & 3 & 4 & 1 & 3.5 & 6 & 2.5 \end{pmatrix} \quad (13)$$

Die ersten sechs Spalten gehören zur Klasse C_{ja}, die letzten drei zur Klasse C_{nein}. In der Matrix W2 wird diesmal festgelegt, daß die Klasse C_{ja} durch zwei Codebook-Vektoren und die Klasse C_{nein} durch einen Codebook-Vektor repräsentiert werden soll. Nach der Trainingsphase (2000 Epochen) enthalten die Zeilen von W1 wieder die ermittelten Codebuch- oder Codebook-Vektoren. Ohne Berücksichtigung der Monotonie sind das die Referenzdatensätze oder Codebook-Vektoren

w1 = (2.0519, 4.3227)
w2 = (4.2409, 2.5944)
w3 = (5.9431, 4.0749) (14)

Diese sind in der entsprechenden Graphik der **Fig.** 6a wieder als O und ⊠ markiert, während die Vektoren der Trainingsmenge mit + und ✱ gekennzeichnet sind. Die Symbole O und + stehen dabei für die Klasse C_{ja}, während die Symbole ⊠ und ✱ die Klasse C_{nein} kennzeichnen. Die vom neuronalen Netz ermittelte Klassengrenze setzt sich hier zusammen aus der Mittelsenkrechten zwischen w1 und w3 und aus der Mittelsenkrechten zwischen w2 und w3. Sie erfüllt die Forderung der Monotonie in positiver und negativer 2-Richtung nicht.

Mit Berücksichtigung der Monotonie nach der neuen Methode werden als Codebook-Vektoren berechnet

w1 = (3.7065, 2.7192)
w2 = (1.9509, 4.3476)
w3 = (6.3209, 5.2759) (15)

die in der **Fig.** 6b in analoger Symbolik zur **Fig.** 6a gezeigt sind.

Die vom neuronalen Netz ermittelte Klassengrenze erfüllt hier die Forderung der Monotonie in positiver und negativer 2-Richtung. Dies kann wieder belegt werden durch Simulationen mit dem Programm "simulvq" MATLAB Neural Network TOOLBOX (a.a.O.).

Als interessant bei der durchgeführten Simulation zeigte sich, daß der ursprünglich als "ja" klassifizierte Trainingsvektor $(5,1)^T$ nun unter Anwendung der Erfindung als "nein" klassifiziert wurde. Ähnlich wurde der ursprünglich als "nein" klassifizierte Trainingsvektor $(5, 3.5)^T$ nun bei einem erfindungsgemäßen neuronalen Netz und entsprechendem Klassifizierungsverfahren als "ja" klassifiziert. Diese Ergebnisse sind aber durchaus erwünscht, da die ursprüngliche Klassifikation dieser beiden Trainingsvektoren ja im Widerspruch zur Forderung dieser Monotonie steht. Gemäß der Erfindung hat somit die vorgegebene, d. h. a priori vorhandene Monotonieeigenschaft Vorzug vor der Klassifizierung der Trainingsvektoren. Durch die Erfindung wird also der Monotonie-Forderung ein höherer Stellenwert eingeräumt als den Trainingsdaten. Dies ist absolut sinnvoll, da die Monotonie-Forderung eine exakte, harte Eigenschaft darstellt, während die Trainingsdaten mit statistischen Meß- und Erhebungsfehlern behaftet sind.

Aus Gründen der einfacheren Darstellung wurde teilweise nur der Fall der Monotonie in einer Richtung behandelt. Das Verfahren läßt sich jedoch unmittelbar verallgemeinern. Wenn etwa die Klasse C monoton in positiver i-Richtung und in negativer j-Richtung ist, so muß als Abstand zwischen $X = (x_1, \ldots, x_n)$ und dem zur Klasse C gehörigen Codebook-Vektor oder Referenzdatensatz $W = (w_1, \ldots, w_n)$ bestimmt werden:

$$d(X,W) = \begin{cases} \sqrt{\sum\limits_{\substack{v \neq i \\ v \neq j}}^{n} (x_v - w_v)^2} \;\ldots\ldots & \text{falls } x_i > w_i \text{ und } x_j < w_j \\ \sqrt{\sum\limits_{v \neq i}^{n} (x_v - w_v)^2} \;\ldots\ldots & \text{falls } x_i > w_i \text{ und } x_j \geq w_j \\ \sqrt{\sum\limits_{v \neq j}^{n} (x_v - w_v)^2} \;\ldots\ldots & \text{falls } x_i \leq w_i \text{ und } x_j < w_j \\ \|X - W\| = \sqrt{\sum\limits_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \leq w_i \text{ und } x_j \geq w_j \end{cases} \quad (16)$$

Entsprechend lassen sich alle Kombinationen behandeln. Es gilt dann wieder eine der weiter oben angegebenen Entwicklung für den praktisch wichtigen Fall zweier Klassen mit gegenläufigen Monotonie-Eigenschaften entsprechende

Aussage.

Für neuronale LVQ-Netze gibt es zahlreiche Anwendungsmöglichkeiten, bei denen sich die Erfindung mit Vorteil einsetzen läßt.

Beispielsweise werden als typische Vertreter von Klassifizierungsthematiken zunächst betriebswirtschaftliche Anwendungen angeführt. Nach N. P. Archer, S. Wang: "Application of the Back Propagation Neural Network Algorithm with Monotonicity Constraints for Two-Group Classification Problems" (Abstract; a.a.O.) besitzen die meisten Klassifikationsaufgaben aus diesem Bereich Monotonie-Eigenschaften. Neben Kaufakt- und Markenwahlmodellen ist hier vor allem die Anwendung von neuronalen Netzen zur Kreditwürdigkeitsprüfung zu nennen.

Weiterer Hintergrund für derartige Anwendungen bei der Klassifikation von Unternehmen mittels künstlichen neuronalen Netzen finden sich in M. Kerling, T. Poddig: "Klassifikation von Unternehmen mittels KNN", in "Neuronale Netze in der Ökonomie" (herausgegeben von H. Rehkugler und H. G. Zimmermann), S. 427 bis 490, Verlag Franz Vahlen, München (1994), und C. Krause: "Kreditwürdigkeitsprüfung mit neuronalen Netzen", Dissertation am Institut für Revisionswesen der Westfälischen Wilhelms-Universität Münster, IDW-Verlag GmbH, Düsseldorf, 1993.

Mit der Anwendung von künstlichen neuronalen Netzen zur Kreditwürdigkeitsprüfung bei Privatkundenkrediten befassen sich T. Lohrbach: "Einsatz von künstlichen neuronalen Netzen für ausgewählte betriebswirtschaftliche Aufgabenstellungen und Vergleich mit konventionellen Lösungsverfahren", Dissertation am Institut für Wirtschaftsinformatik der Universität Göttingen, Unitext-Verlag, Göttingen, 1994, und A. Schmidt-von-Rhein, H. Rehkugler: "KNN zur Kreditwürdigkeitsprüfung bei Privatkundenkrediten", in "Neuronale Netze in der Ökonomie" (herausgegeben von H. Rehkugler und H. G. Zimmermann), 491 bis 545, Verlag Franz Vahlen, München (1994).

Die Netztypen MLP (Multilayer-Perceptron) und LVQ schlagen nach A. Schmidt-von-Rhein, H. Rehkugler: "KNN zur Kreditwürdigkeitsprüfung bei Privatkundenkrediten" (S. 541, a.a.O.) in diesen Bereichen die lineare multivariante Diskriminanzanalyse (MDA), welches das konkurrierende Standardverfahren aus der Statistik darstellt (vgl. etwa L. Fahrmeir, W. Häussler, G. Tutz: "Diskriminanzanalyse", in "Multivariante statistische Verfahren" (herausgegeben von L. Fahrmeir und A. Hamerle), S. 301–370, Walter de Gruyter, Berlin (1984)). Die neuronalen Netze haben gegenüber MDA den Vorteil, daß weitgehend beliebige nichtlineare Zusammenhänge abbildbar sind. Sie machen auch geringere theoretische Voraussetzungen über ihre Anwendbarkeit. Der Netztyp LVQ hat gegenüber MLP jedenfalls den Vorteil, daß die Resultate des ersteren besser interpretierbar sind. Diese Tatsache ist für die Akzeptanz des Verfahrens im betriebswirtschaftlichen Umfeld von Bedeutung.

Die Offenbarungen der vorstehend angegebenen Literaturquellen sind durch die Bezugnahme hiermit vollumfänglich in die vorliegenden Unterlagen aufgenommen.

Über die genannten wirtschaftlichen Anwendungen des neuronalen LVQ-Netzes und des darauf laufenden Klassifizierungsverfahrens nach der Erfindung gibt es aber noch weitere Nutzungsmöglichkeiten, wie beispielsweise bei der Behandlung von Steuerproblemen. Aber auch auf allen anderen Gebieten läßt sich die Erfindung in Form ihres neuronalen LVQ-Netzes und des entsprechenden Klassifizierungsverfahrens mit Vorteil einsetzen. Lediglich exemplarisch seien hier die Biologie, Neurobiologie, Bioinformatik, Medizin, Neurophysiologie, Psychologie, Elektrotechnik, Physik, Mathematik, Informatik und Künstliche Intelligenz als Gebiete genannt, auf denen zur Lösung von vielfältigen Klassifikationsaufgaben, wie z. B. zur Musterkennung, das erfindungsgemäße neuronale LVQ-Netz und das damit verbundene Klassifizierungsverfahren mit Vorteil eingesetzt werden können, wie beispielsweise in "Simulation Neuronaler Netze" von Andreas Zell (S. 24 und S. 496 bis 574 a.a.O.) und auch in "Neuronale Netze" von Rüdiger Brause (a.a.O.) mit zahlreichen konkreten Angaben beschreiben ist. Lediglich exemplarisch zur Verdeutlichung der möglichen Vielfalt der Anwendung der Erfindung kann sie bei der Identifikation von Meeresverschmutzern, wie z. B. durch Ölablaß aus Tankern, (siehe z. B. Y. T. Chien und T. J. Killeen: "Computer and Statistical Considerations for Oil Spill Identification" in "Handbook of Statistics", Vol. 2 von Krishnaiah und Kanel (Herausgeber), North-Holland Publishing Company, 1982, S. 651 bis 671, mit weiteren Nachweisen) oder beim Aufspüren von Geldwäschetransaktionen eingesetzt werden.

Die Erfindung sowohl in Form eines neuronalen Netzes zur lernenden Vektorquantisierung, als auch durch das Klassifizierungsverfahren mittels eines solchen neuronalen Netzes, kann vielfältig implementiert werden. So eigenen sich neben digitalen oder analogen VLSI-Chips, Neurocomputern, insbesondere aus Standardbausteinen, wie Signalprozessoren, SIMD-Parallelrechnern, MIMD-(Parallel-)Rechnern, Zusatzkarten, wie z. B. Koprozessor-Boards für neuronale Netze, insbesondere mit mehreren Koprozessoren für vorzugsweise hohe Gleitkommaleistung, für Workstations oder PCs, VLSI-Neurocomputern, insbesondere mit Spezialhardware, für eine größere Klasse von konnektionistischen Modellen, auch zwischenzeitlich verfügbare optische Neurocomputersysteme.

Ferner ermöglicht eine Adaption von Workstations oder PCs mit geeigneter Simulationssoftware für ein neuronales Netz zur lernenden Vektorquantisierung auch deren Einsatz zur Erstellung eines neuronalen Netzes und zur Durchführung eines Klassifizierungsverfahrens damit jeweils mit den erfindungsgemäßen Merkmalen. Dabei stellt das durch eine Software zusammen mit Workstations oder PCs geschaffene neuronale Netz insbesondere mit der implementierten vorliegenden Erfindung keine reine Computer-Software als solche dar, sondern bildet zusammen mit der Hardware eine technisch untrennbare Einheit, worauf u. a. auch in dem Buch "Neuronale Netze" von Rüdiger Brause (insbesondere S. 38, a.a.O.) hingewiesen ist.

Hinsichtlich der Realisierungsmöglichkeiten und -formen eines neuronalen Netzes wird auf die zahlreichen und detaillierten Angaben hierzu in den Büchern "Simulation Neuronaler Netze" von Andreas Zell (a.a.O.), "Neuronale Netze" von Rüdiger Brause (a.a.O.) und "Self-Organizing Maps" von T. Kohonen (a.a.O.) Bezug genommen und dadurch der Offenbarungsgehalt dieser Veröffentlichungen insbesondere bezüglich der Bau- und Realisierungsgestaltungen, aber auch im übrigen, vollumfänglich in diese Beschreibung aufgenommen. Grundsätzlich sind alle in den genannten Literaturquellen angegebenen neuronalen Netze zur Implementierung der lernenden Vektorquantisierung geeignet.

Die Erfindung kann allgemein auch noch in anderen Ausführungsformen und Formulierungen zum Ausdruck gebracht werden.

So wird gemäß der vorliegenden Erfindung auch ein künstliches neuronales Netz (KNN) zur lernenden Vektorquantisierung (LVQ) geschaffen, das wenigstens eine Schicht aktiver Neuronen, denen Referenzvektoren zugeordnet und die in

Ausgabeklassen eingeteilt sind, und Eingabeeinrichtungen für Eingabevektoren enthält, wobei das KNN dazu ausgelegt ist, durch Vergleich eines Eingabevektors parallel mit allen Referenzvektoren dasjenige Neuron zu aktivieren, dessen Referenzvektor dem Eingabevektor am ähnlichsten ist. Dabei ist vorrichtungsmäßig und wird verfahrensmäßig im Rahmen der Erfindung bei der Ähnlichkeitsbestimmung zwischen den Eingabevektoren $X = (x_1, \ldots, x_n)$ $(n \in N)$ und den Referenzvektoren $W = (w_1, \ldots, w_n)$ $(m \in N;\ m \geq n)$ bei zumindest einer Komponente $x_i (i \in (1, \ldots, m))$ der Eingabevektoren X eine vorbekannte Eigenschaft, wie z. B. Monotonie oder Symmetrie, zu Grunde gelegt.

Die Erfindung kann aber auch so betrachtet werden, daß ein künstliches neuronales Netz zur lernenden Vektorquantisierung Eingabeeinrichtungen für n-dimensionale Eingabedatensätze $(n \in N)$ und wenigstens eine verdeckte Neuronenschicht enthält, deren Neuronen Referenzdatensatz-Speichereinrichtungen zum Speichern m-dimensionaler Referenzdatensätze $(m \in N;\ m \geq n)$ und Verarbeitungseinrichtungen zum Ermitteln desjenigen Referenzdatensatzes bilden, der einem Eingabedatensatz am ähnlichsten ist, und daß die Verarbeitungseinrichtungen dazu ausgelegt sind, bei der Ähnlichkeitsbestimmung zwischen den Eingabedatensätzen $X = (x_1, \ldots, x_n)$ und den Referenzdatensätzen $W = (w_1, \ldots, w_n)$ bei zumindest einer Komponente x_i $(i \in (1 \text{ bis } m))$ der Eingabedatensätze X eine vorbekannte Eigenschaft, wie z. B. Monotonie oder Symmetrie, zu Grunde zu legen. Das Verfahren gestaltet sich entsprechend.

In einer Weiterbildung davon kann die von den Verarbeitungseinrichtungen bei der Bestimmung des Abstandes als Maß für die Ähnlichkeit zwischen den Eingabedatensätzen X und den Referenzdatensätzen W zu Grunde gelegte Monotonieeigenschaft wenigstens in positiver oder negativer i-Richtung der i-ten Komponente x_i der Eingabedatensätze X vorhanden sein.

Eine weitere Sichtweise der Erfindung kann durch ein Multiprozessorsystem mit einer Mehrzahl paralleler Prozessoreinheiten, denen jeweils ein insbesondere lokaler Speicher zum Speichern eines Referenzdatensatzes zugeordnet ist, und Eingabeeinrichtungen für einen Eingabedatensatz zum Ausdruck gebracht werden, der durch paralleles vergleichendes Ermitteln der Ähnlichkeit mit allen Referenzdatensätzen entsprechend dem ähnlichsten Referenzdatensatz klassifizierbar ist. Bei der Ähnlichkeitsbestimmung zwischen den Eingabevektoren X und den Referenzvektoren W ist bei der Vorrichtung und wird bei dem Klassifizierungsverfahren jeweils nach der Erfindung bei zumindest einer Komponente x_i der Eingabevektoren X eine vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie oder Symmetrie, zu Grunde gelegt.

Die in dieser Beschreibung angegebene oder zitierte Literatur gehört genauso zum Inhalt und Offenbarungsgehalt dieser Unterlagen, wie wenn sie vollständig darin wiedergegeben wäre.

Patentansprüche

1. Neuronales Netz zur lernenden Vektorquantisierung, mit Eingabeeinrichtungen für Eingabedatensätze und wenigstens einer Schicht aktiver Neuronen, die mit den Eingabeeinrichtungen verbunden und denen Referenzdatensätze zugeordnet sind, um zur Klassifizierung eines Eingabedatensatzes durch Vergleich mit allen Referenzdatensätzen dasjenige Neuron der Schicht aktiver Neuronen, dessen Referenzdatensatz dem Eingabedatensatz am ähnlichsten ist, zu einer Ausgabe an Ausgabeeinrichtungen zu veranlassen, **dadurch gekennzeichnet**, daß bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und den Referenzdatensätzen (W) bei zumindest einer seiner Komponenten (xi) eine vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, zu Grunde gelegt ist.
2. Neuronales Netz nach Anspruch 1, dadurch gekennzeichnet, daß die Eingabeeinrichtungen eine Eingabeneuronenschicht enthalten, deren Eingabeneuronen jeweils vollständig mit jedem Neuron der Schicht aktiver Neuronen verbunden sind.
3. Neuronales Netz nach Anspruch 2, dadurch gekennzeichnet, daß jedem Eingabeneuron der Eingabeneuronenschicht eine Komponente (x_i) eines Eingabedatensatzes (X) zugeordnet ist.
4. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Neuronen einer Schicht aktiver Neuronen in Ausgabeklassen eingeteilt sind.
5. Neuronales Netz nach Anspruch 4, dadurch gekennzeichnet, daß jeder Ausgabeklasse wenigstens ein Neuron und insbesondere eine Vielzahl von Neuronen einer Schicht aktiver Neuronen zugeordnet ist.
6. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Ausgabeeinrichtungen eine Ausgabeneuronenschicht enthalten, deren Ausgabeneuronen jeweils eine Ausgabeklasse repräsentieren und in Abhängigkeit von dem ermittelten Neuron der vorhergehenden Schicht aktiver Neuronen aktivierbar sind.
7. Neuronales Netz nach einem der Ansprüche 4 bis 6, dadurch gekennzeichnet, daß die vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, für zumindest eine Ausgabeklasse gilt.
8. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß es zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) dazu ausgelegt ist, zu denjenigen Referenzdatensätzen (W), deren Klasse wenigstens eine vorgegebene Eigenschaft, wie z. B. eine Monotonieeigenschaft, besitzt, eine Punktmenge zu bilden, die aus einem Referenzdatensatz und denjenigen Datensätzen gebildet ist, die von dem Referenzdatensatz aus in der/den Richtung(en) der einen oder mehreren vorgegebenen Eigenschaft(en), wie z. B. Monotonieeigenschaft(en), liegen, und anhand derjenigen Punktmenge, von der der Eingabedatensatz (X) den geringsten Abstand, insbesondere den geringsten euklidischen Abstand, aufweist, den zugehörigen Referenzdatensatz (W) als denjenigen festzulegen, der dem Eingabedatensatz (X) am ähnlichsten ist.
9. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß zur Klassifizierung eines Eingabedatensatzes (X) bei zumindest einer seiner Komponenten (x_i) wenigstens ein Gültigkeitsbereich einer vorbekannten Eigenschaft der Klassifizierung, wie z. B. einer Monotonieeigenschaft, bestimmt ist.
10. Neuronales Netz nach Anspruch 9, dadurch gekennzeichnet, daß es dazu ausgelegt ist, bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) die der vorgegebenen Eigenschaft der Klassifizierung entsprechende Komponente (x_i) des Eingabedatensatzes (X) unberücksichtigt zu las-

sen, wenn der Eingabedatensatz (X) in dem entsprechenden Gültigkeitsbereich der vorbekannten Eigenschaft der Klassifizierung liegt und gegenüber dem Referenzdatensatz (W) eine dieser Eigenschaft entsprechende Lage hat.

11. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß es zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) dazu ausgelegt ist, denjenigen Referenzdatensatz (W) zu bestimmen, der den geringsten Abstand, insbesondere den geringsten euklidischen Abstand von dem Eingabedatensatz (X) aufweist.

12. Neuronales Netz nach Anspruch 9 oder 10 jeweils in Verbindung mit Anspruch 11, dadurch gekennzeichnet, daß es dazu ausgelegt ist, den Abstand zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) zu bestimmen, indem zu allen Referenzdatensätzen (W), welche zu Klassen mit einer oder mehreren vorgegebenen Eigenschaften, wie z. B. Monotonieeigenschaft(en), gehören, ausgehend von jedem dieser Referenzdatensätze (W) in Richtung der einen oder mehreren vorgegebenen Eigenschaft(en), wie z. B. Monotonieeigenschaft(en), eine zugehörige Punktmenge gebildet wird und dann der Abstand des Eingabedatensatzes (X) von dieser Punktmenge bestimmt wird, soweit diese in dem/den Gültigkeitsbereich(en) der vorgegebenen Eigenschaft(en) liegt.

13. Neuronales Netz nach einem der Ansprüche 9 bis 12, dadurch gekennzeichnet, daß ein Gültigkeitsbereich einer vorbekannten Eigenschaft der Klassifizierung, wie z. B. einer Monotonieeigenschaft, für eine Komponente (x_i) eines Eingabedatensatzes (X) die positive oder negative i-Richtung der Komponente (x_i) des Eingabedatensatzes (X) ist.

14. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß es dazu ausgelegt ist, zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) den jeweiligen Abstand zwischen dem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) bei einer Monotonieeigenschaft einer Komponente (x_i) der Eingabedatensätze (X) in positiver i-Richtung gemäß

$$d(X,W) = \begin{cases} \sqrt{\sum\limits_{\substack{v=1 \\ v\neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i > w_i \\ \|X - W\| = \sqrt{\sum\limits_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \leq w_i \end{cases}$$

bestimmt wird, wobei $X = (x_1, \ldots, x_n)$ und $W = (w_1, \ldots, w_n)$.

15. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß es dazu ausgelegt ist, zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) den jeweiligen Abstand zwischen dem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) bei einer Monotonieeigenschaft einer Komponente (x_i) der Eingabedatensätze (X) in negativer i-Richtung gemäß

$$d(X,W) = \begin{cases} \sqrt{\sum\limits_{\substack{v=1 \\ v\neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i < w_i \\ \|X - W\| = \sqrt{\sum\limits_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \geq w_i \end{cases}$$

bestimmt wird, wobei $X = (x_1, \ldots, x_n)$ und $W = (w_1, \ldots, w_n)$.

16. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Neuronen der Schicht aktiver Neuronen jeweils eine Prozessoreinheit und einen lokalen Speicher zur Speicherung eines Referenzdatensatzes (W) enthalten, daß die Schicht durch eine Parallelanordnung der Prozessoreinheiten mit ihren zugeordneten Speichern gebildet ist, und daß jede Prozessoreinheit zur Durchführung des Vergleichs des eingegebenen Eingabedatensatzes (X) mit dem Referenzdatensatz (W) in dem zugehörigen Speicher ausgelegt ist.

17. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Schicht aktiver Neuronen zum parallelen Vergleich eines Eingabedatensatzes (X) mit allen Referenzdatensätzen (W) ausgelegt ist.

18. Neuronales Netz nach einem der Ansprüche 1 bis 15, dadurch gekennzeichnet, daß die Neuronen der Schicht aktiver Neuronen zumindest durch eine zentrale Verarbeitungseinheit und einen zentralen Speicher zum Speichern aller Referenzdatensatzes (W) gebildet sind, und daß die zentrale Verarbeitungseinheit zum Durchführen des Vergleichs des eingegebenen Eingabedatensatzes (X) mit jedem Referenzdatensatz (W) in dem zentralen Speicher in paralleler oder bezüglich jedes Referenzdatensatzes (W) unmittelbar serieller oder bezüglich jeder Komponente (x_i) des Eingabedatensatzes (X) unmittelbar serieller Weise ausgelegt ist.

19. Neuronales Netz nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Eingabedatensätze (X) durch Eingabevektoren und die Referenzdatensätze (W) durch Referenzvektoren gebildet sind.

20. Verfahren zur Klassifizierung von Eingabedatensätzen mit einem neuronalen Netz zur lernenden Vektorquantisierung, wobei Eingabedatensätze über Eingabeeinrichtungen einer Ähnlichkeitsbestimmung bezüglich Referenzdatensätzen unterzogen werden, die jeweils einem Neuron einer Schicht aktiver Neuronen zugeordnet sind, und zur Klassifizierung durch Vergleichen eines Eingabedatensatzes mit allen Referenzdatensätzen dasjenige Neuron der Schicht aktiver Neuronen, dessen Referenzdatensatz dem Eingabedatensatz am ähnlichsten ist, zu einer Ausgabe an Ausgabeeinrichtungen veranlaßt wird, dadurch gekennzeichnet, daß bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und den Referenzdatensätzen (W) bei zumindest einer seiner Komponenten (x_i) eine vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, zu

Grunde gelegt wird.

21. Verfahren nach Anspruch 20, dadurch gekennzeichnet, daß jeweils eine Komponente (x_i) eines Eingabedatensatzes (X) einem Eingabeneuron zugeordnet wird, das Teil einer Eingabeneuronenschicht ist, die in den Eingabeeinrichtungen enthalten ist, wobei die Eingabeneuronen jeweils vollständig mit jedem Neuron der Schicht aktiver Neuronen verbunden sind.

22. Verfahren nach Anspruch 20 oder 21, dadurch gekennzeichnet, daß jedem Eingabeneuron der Eingabeneuronenschicht eine Komponente (x_i) eines Eingabedatensatzes (X) zugeordnet wird.

23. Verfahren nach einem der Ansprüche 20 bis 22, dadurch gekennzeichnet, daß die Neuronen einer Schicht aktiver Neuronen in Ausgabeklassen eingeteilt werden.

24. Verfahren nach Anspruch 23, dadurch gekennzeichnet, daß jeder Ausgabeklasse wenigstens ein Neuron und insbesondere eine Vielzahl von Neuronen einer Schicht aktiver Neuronen zugeordnet wird.

25. Verfahren nach einem der Ansprüche 20 bis 24, dadurch gekennzeichnet, daß die Ausgabeeinrichtungen eine Ausgabeneuronenschicht enthalten, deren Ausgabeneuronen jeweils eine Ausgabeklasse repräsentieren und in Abhängigkeit von dem ermittelten Neuron der vorhergehenden Schicht aktiver Neuronen aktiviert werden.

26. Verfahren nach einem der Ansprüche 23 bis 25, dadurch gekennzeichnet, daß die vorbekannte Eigenschaft der Klassifizierung, wie z. B. Monotonie in wenigstens einer Richtung oder Symmetrie, für zumindest eine Ausgabeklasse verwendet wird.

27. Verfahren nach einem der Ansprüche 20 bis 26, dadurch gekennzeichnet, daß zur Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) zu denjenigen Referenzdatensätzen (W), deren Klasse wenigstens eine vorgegebene Eigenschaft, wie z. B. eine Monotonieeigenschaft, besitzt, eine Punktmenge gebildet wird, die aus einem Referenzdatensatz und denjenigen Datensätzen gebildet ist, die von dem Referenzdatensatz aus in der/den Richtung(en) der einen oder mehreren vorgegebenen Eigenschaft(en), wie z. B. Monotonieeigenschaft(en), liegen, und anhand derjenigen Punktmenge, von der der Eingabedatensatz (X) den geringsten Abstand, insbesondere den geringsten euklidischen Abstand, aufweist, der zugehörige Referenzdatensatz (W) als derjenige festgelegt wird, der dem Eingabedatensatz (X) am ähnlichsten ist.

28. Verfahren nach einem der Ansprüche 20 bis 27, dadurch gekennzeichnet, daß zur Klassifizierung eines Eingabedatensatzes (X) bei zumindest einer seiner Komponenten (x_i) wenigstens ein Gültigkeitsbereich einer vorbekannten Eigenschaft der Klassifizierung, wie z. B. einer Monotonieeigenschaft, bestimmt ist.

29. Verfahren nach Anspruch 28, dadurch gekennzeichnet, daß bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) die der vorgegebenen Eigenschaft der Klassifizierung entsprechende Komponente (x_i) des Eingabedatensatzes (X) unberücksichtigt bleibt, wenn der Eingabedatensatz (X) in dem entsprechenden Gültigkeitsbereich der vorbekannten Eigenschaft der Klassifizierung liegt und gegenüber dem Referenzdatensatz (W) eine dieser Eigenschaft entsprechende Lage hat.

30. Verfahren nach einem der Ansprüche 20 bis 29, dadurch gekennzeichnet, daß bei der Ähnlichkeitsbestimmung zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) derjenige Referenzdatensatz (W) bestimmt wird, der den geringsten Abstand, insbesondere den geringsten euklidischen Abstand von dem Eingabedatensatz (X) aufweist.

31. Verfahren nach Anspruch 28 oder 29 jeweils in Verbindung mit Anspruch 30, dadurch gekennzeichnet, daß der Abstand zwischen einem Eingabedatensatz (X) und jedem der Referenzdatensätze (W) bestimmt wird, indem zu allen Referenzdatensätzen (W), welche zu Klassen mit einer oder mehreren vorgegebenen Eigenschaften, wie z. B. Monotonieeigenschaft(en), gehören, ausgehend von jedem dieser Referenzdatensätze (W) in Richtung der einen oder mehreren vorgegebenen Eigenschaft(en), wie z. B. Monotonieeigenschaft(en), eine zugehörige Punktmenge gebildet wird und dann der Abstand des Eingabedatensatzes (X) von dieser Punktmenge bestimmt wird, soweit diese in dem/den Gültigkeitsbereich(en) der vorgegebenen Eigenschaft(en) liegt.

32. Verfahren nach einem der Ansprüche 28 bis 31, dadurch gekennzeichnet, daß ein Gültigkeitsbereich einer vorbekannten Eigenschaft der Klassifizierung, wie z. B. einer Monotonieeigenschaft, für eine Komponente (x_i) eines Eingabedatensatzes (X) die positive oder negative i-Richtung der Komponente (xi) des Eingabedatensatzes (X) ist.

33. Verfahren nach einem der Ansprüche 20 bis 32, dadurch gekennzeichnet, daß es dazu ausgelegt ist, den Abstand zwischen einem Eingabedatensatz (X) und den Referenzdatensätzen (W) bei einer Monotonieeigenschaft einer Komponente (x_i) der Eingabedatensätze (X) in positiver i-Richtung gemäß

$$d(X,W) = \begin{cases} \sqrt{\sum_{\substack{v=1 \\ v\neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i > w_i \\ \|X - W\| = \sqrt{\sum_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \leq w_i \end{cases}$$

bestimmt wird, wobei $X = (x_i, \ldots, x_n)$ und $W = (w_i, \ldots, w_n)$.

34. Verfahren nach einem der Ansprüche 20 bis 33, dadurch gekennzeichnet, daß es dazu ausgelegt ist, den Abstand zwischen den Eingabedatensätzen (X) und den Referenzdatensätzen (W) bei einer Monotonieeigenschaft einer Komponente (x_i) der Eingabedatensätze (X) in negativer i-Richtung gemäß

$$d(X,W) = \begin{cases} \sqrt{\sum_{\substack{v=1 \\ v \neq i}}^{n} (x_v - w_v)^2} & \text{falls } x_i < w_i \\ \|X - W\| = \sqrt{\sum_{v=1}^{n} (x_v - w_v)^2} & \text{falls } x_i \geq w_i \end{cases}$$

bestimmt wird, wobei $X = (x_1, \ldots, x_n)$ und $W = (w_1, \ldots, w_n)$.

35. Verfahren nach einem der Ansprüche 20 bis 34, dadurch gekennzeichnet, daß das Vergleichen eines eingegebenen Eingabedatensatzes (X) mit allen Referenzdatensätzen (W) parallel in den Neuronen der Schicht aktiver Neuronen durchgeführt wird, die jeweils eine Prozessoreinheit und einen lokalen Speicher zur Speicherung eines Referenzdatensatzes (W) enthalten und in der Schicht aktiver Neuronen parallel angeordnet sind.

36. Verfahren nach einem der Ansprüche 20 bis 35, dadurch gekennzeichnet, daß ein Eingabedatensatz (X) parallel mit allen Referenzdatensätzen (W) verglichen wird, die den Neuronen der Schicht aktiver Neuronen zugeordnet sind.

37. Verfahren nach einem der Ansprüche 20 bis 34, dadurch gekennzeichnet, daß das Vergleichen des eingegebenen Eingabedatensatzes (X) mit jedem Referenzdatensatz (W) in paralleler oder bezüglich jedes Referenzdatensatzes (W) unmittelbar serieller oder bezüglich jeder Komponente (x_i) des Eingabedatensatzes (X) unmittelbar serieller Weise in einer zentralen Verarbeitungseinheit durchgeführt wird, wobei die Neuronen der Schicht aktiver Neuronen zumindest durch die zentrale Verarbeitungseinheit und einen zentralen Speicher zum Speichern aller Referenzdatensatzes (W) gebildet sind.

38. Verfahren nach einem der Ansprüche 20 bis 37, dadurch gekennzeichnet, daß die Eingabedatensätze (X) durch Eingabevektoren und die Referenzdatensätze (W) durch Referenzvektoren gebildet sind.

Hierzu 6 Seite(n) Zeichnungen

Nummer: DE 197 34 735 A1
Int. Cl.[6]: G 06 F 15/18
Offenlegungstag: 18. Februar 1999

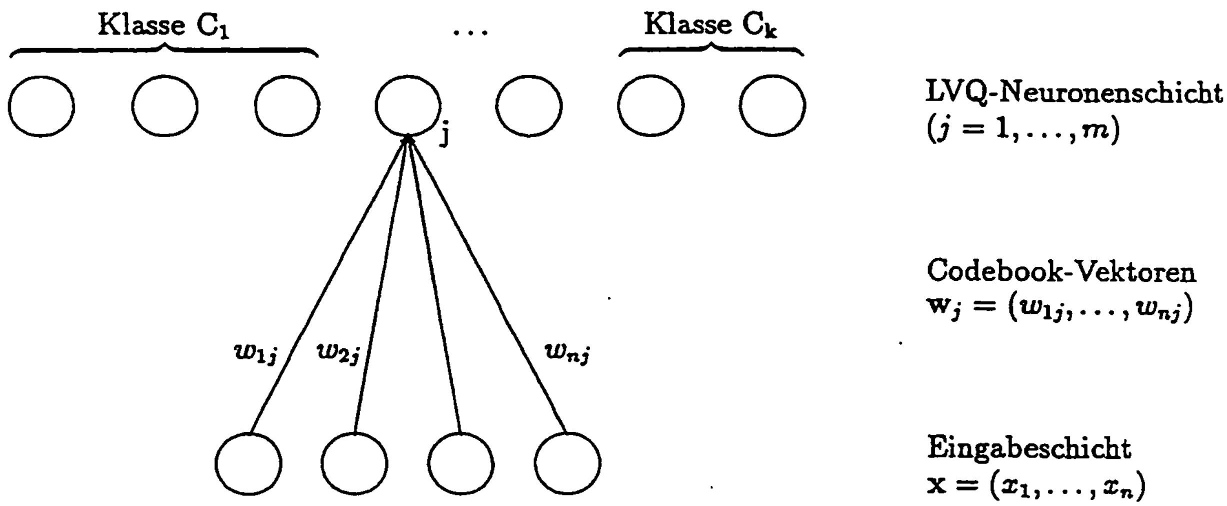
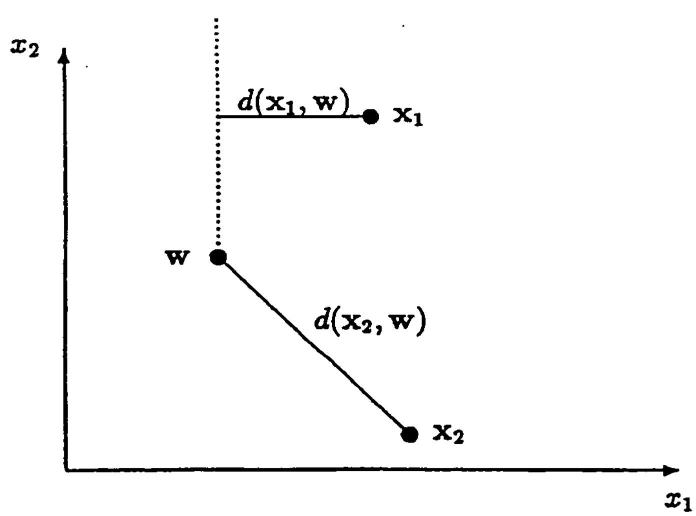


Fig. 1

Fig. 2



Nummer: DE 197 34 735 A1
Int. Cl.[6]: G 06 F 15/18
Offenlegungstag: 18. Februar 1999

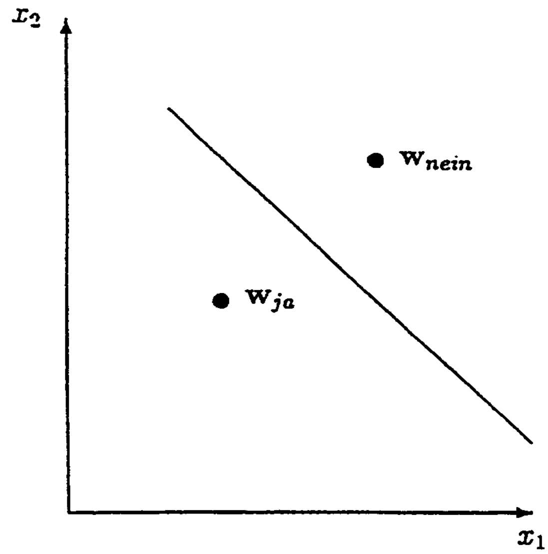


Fig. 3

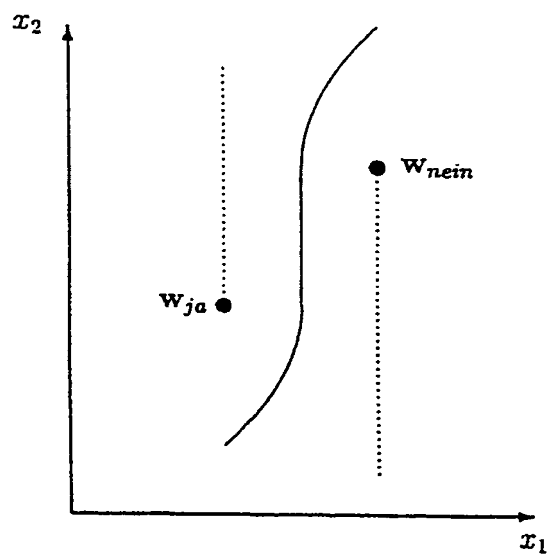


Fig. 4

Nummer: DE 197 34 735 A1
Int. Cl.^{6}: G 06 F 15/18
Offenlegungstag: 18. Februar 1999

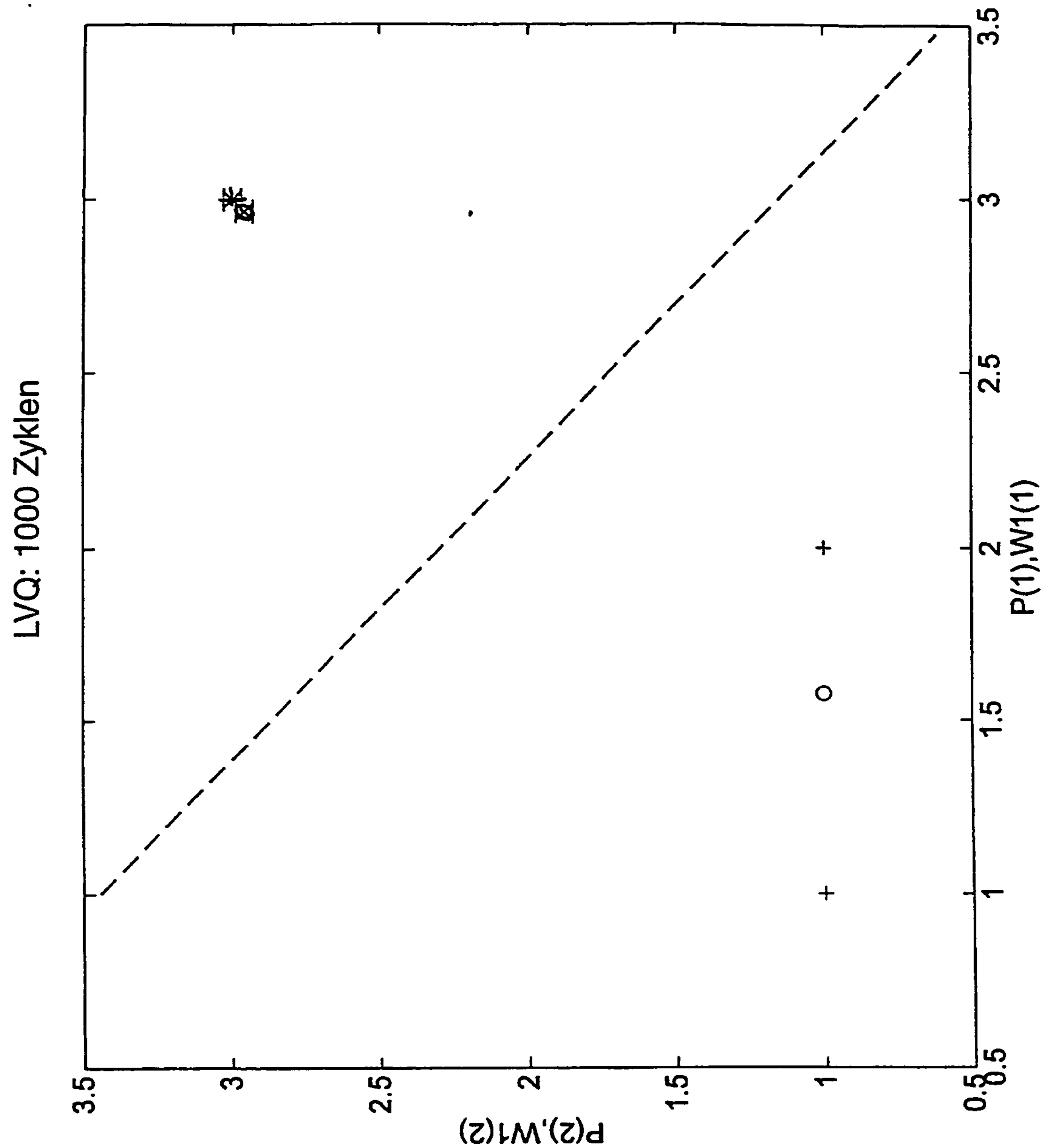


Fig. 5a

Nummer: DE 197 34 735 A1
Int. Cl.^{6}: G 06 F 15/18
Offenlegungstag: 18. Februar 1999

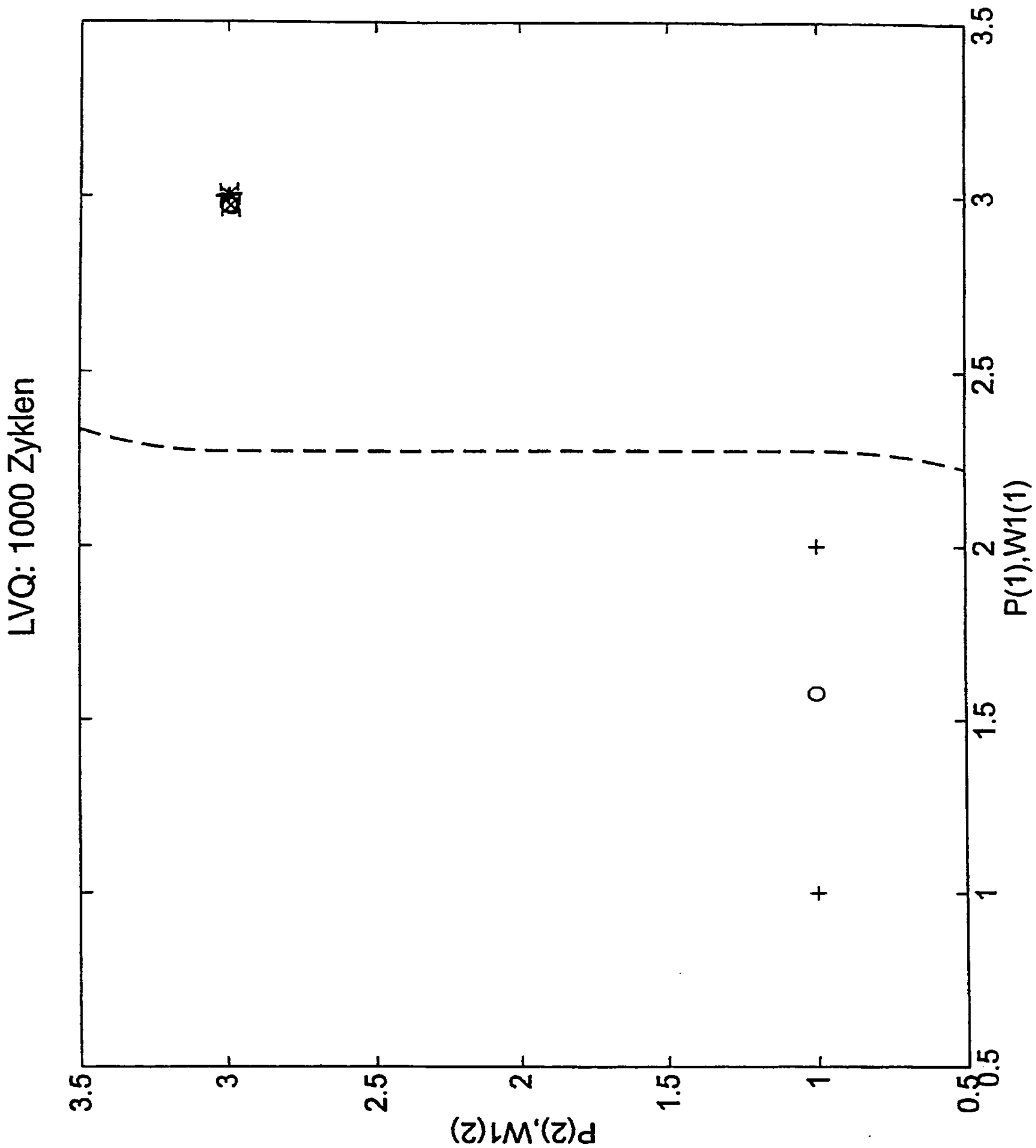


Fig. 5b

Nummer: DE 197 34 735 A1
Int. Cl.[6]: G 06 F 15/18
Offenlegungstag: 18. Februar 1999

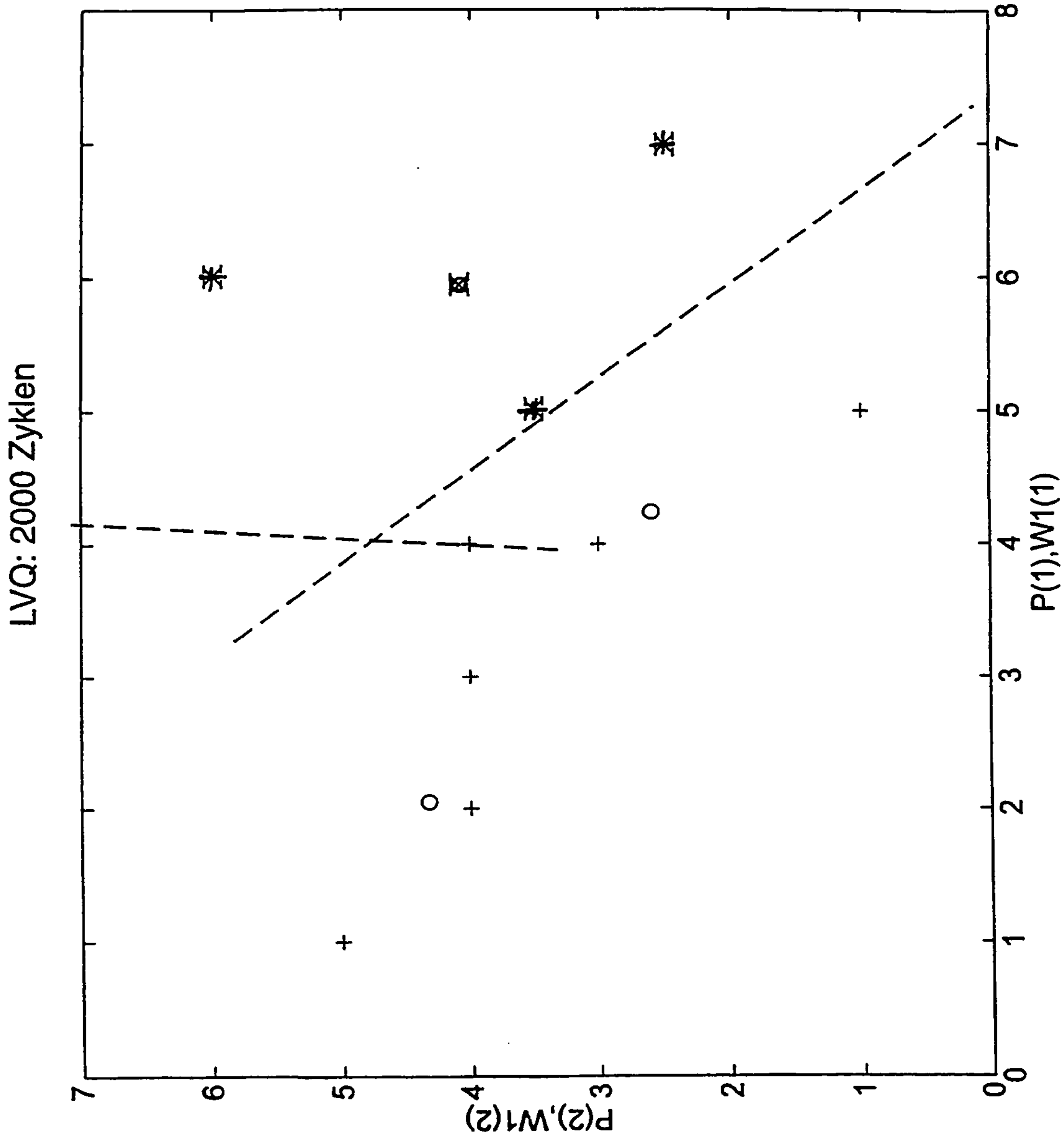


Fig. 6a

Nummer: DE 197 34 735 A1
Int. Cl.[6]: G 06 F 15/18
Offenlegungstag: 18. Februar 1999

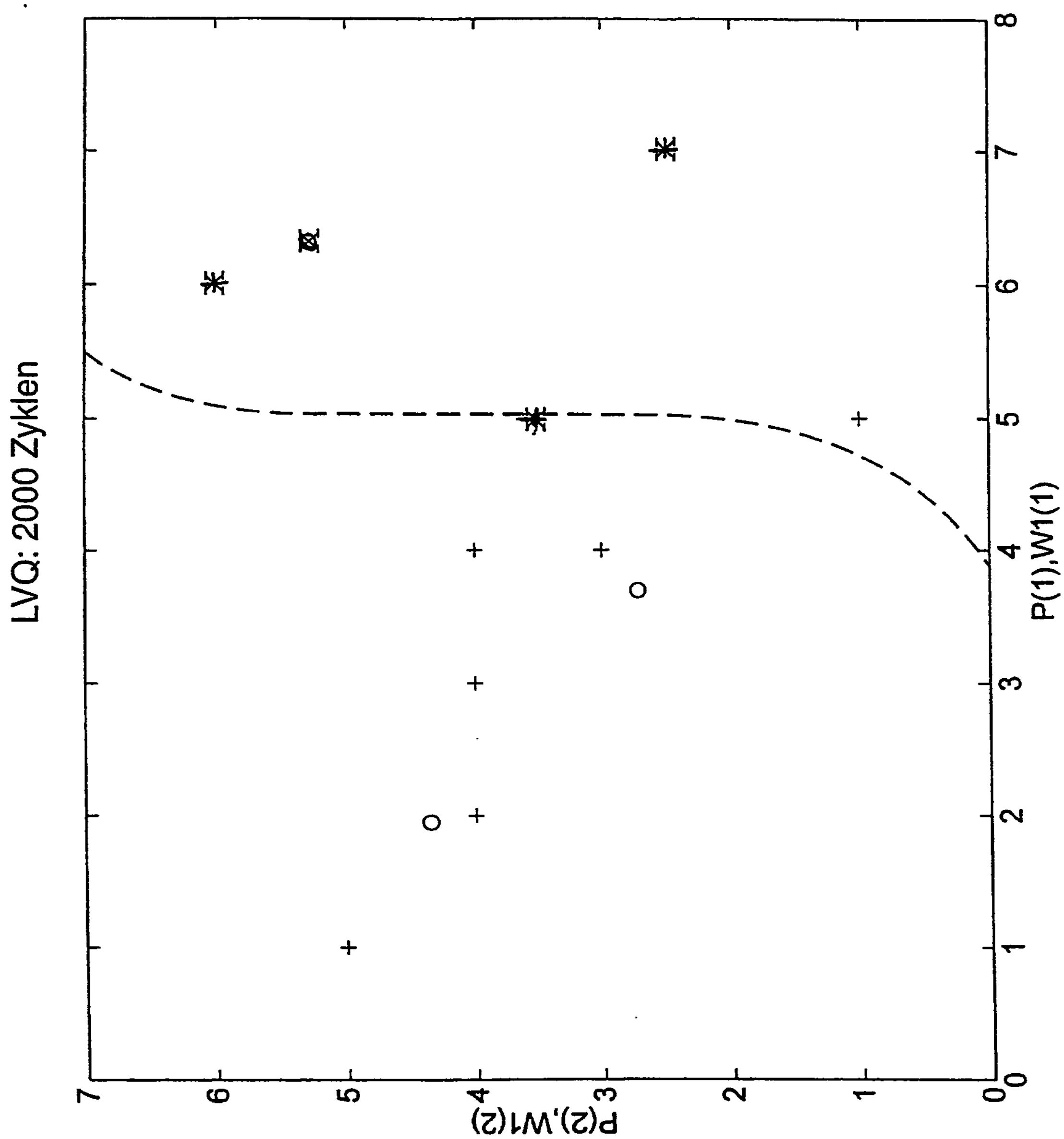


Fig. 6b